International Journal of Advanced Science and Technology
Vol.96 (2016), pp.31-42
http://dx.doi.org/10.14257/ijast.2016.96.04

Proxy based Intelligent Acknowledgment Technique in Multihop Wireless Network

Vani Chawla

PhD Scholar, Department of Electronics and Communication,
AFSET, Faridabad, India
vanichawlaphd@gmail.com

Abstract

Due to the frequently changing topology and high mobility of nodes in multihop wireless network, there may be chances of congestion which results in loss of TCP sequence and degrades the network performance. In existing congestion control works, hop by hop detection incurs huge overhead. Moreover, the waiting time of the source node in obtaining the acknowledgement from the receiver, will be high when the number of hops increases. To overcome these issues, this paper proposes a Proxy based intelligent acknowledgment technique for multihop wireless network. In this technique, proxy nodes are selected along the transmission path, which determine any missing TCP sequence. In this way, packet loss is monitored and retransmission is requested from the source node, thus, ensuring proficient data transmission in the network. Simulation results show that the proposed technique reduces the delay and improves the throughput.

Keywords: *Cross-layer, TCP, Multihop networks, Acknowledgement, Proxy, Link layer*

1. Introduction

1.1. Multihop Wireless Network

A multihop wireless network consists of numerous wireless nodes which actively involve in data transmission by forwarding the packets to the respective destination. When compared to the conventional networks, there are many advantages in using multihop wireless network such as ability to increase the network range, enhance network connectivity, *etc*. Also, packet forwarding through several smaller links is more efficient in terms of lower power consumption when compared to the lengthier links. The multihop wireless networks also facilitate high data rate which in turn provides increased throughput and link utilization. As the density of the multihop wireless network increases the network becomes more dynamic and adaptable due to the increased availability of forwarding paths [1].

In the multihop wireless networks, the source and destination nodes are connected through several wireless hops. The communication between any two intermediate nodes is direct and is performed in a self organized form. One of the most basic issues observed in multihop wireless network is the degradation the network performance, with the increase in the hop count [2]. For instance, in multihop wireless network with omni directional transmission range, a change in hop count from one to two will reduce the data throughput due to wireless interference [3]. Moreover, the channel availability is limited within the given transmission range. So, in case of buffer overflow and link layer contention, the TCP segments will be dropped [4]. As a result of link layer contention, the packets being transmitted gets dropped off, indicating the fact that network is getting overloaded or congested. With the increase in the network load, the packet drop probability also increases, which in turn collapses the network flow [5].

1.2. Issues of TCP in Multihop Networks

The major issue of TCP in handling the congestion is, it cannot differentiate between congestive and non-congestive packet losses. In multi hop wireless networks, TCP has to handle the challenges such as lossy channels, hidden and exposed nodes, path irregularity, network partitions and limitations of power [11]. Improving TCP performance becomes challenging due to large bandwidth and delay variation, high packet loss and recurrent connectivity [12]. TCP recovers packet losses via retransmission technique as it is an end-to-end reliable protocol. TCP depends on acknowledgements from receiver for ensuring reliability [13].

1.3. Problem Identification

In our previous works [2, 14], an improved TCP with intelligent acknowledgment (TCP-IA) technique is proposed for multi-hop wireless networks. In this technique, the sender initiates the transmission of TCP data packet to receiver. The bandwidth and delay for the transmitted packet is computed at the link layer and these values are added to MAC header field. For every k received data packets, the receiver replies with single acknowledgement (ACK). The delay management is performed through window of size k at the receiver that limits the maximum number of ACKs to be delayed. When received data packet has some out-of-order packets or it has minimum bandwidth and maximum delay, the delaying window on the receiver should be reduced in order to avoid performance degradation. Based on the ACK from the receiver, the sender adjusts its congestion window size.

But in this technique, the bandwidth and delay estimation is done at hop by hop basis, which will incur additional overhead. Moreover, the source has to wait for the ACK from the receiver for each transmission which will increase the waiting time when the number of hops increases.

As an extension to the previous work, a proxy based intelligent acknowledgment technique is developed for multi-hop networks.

2. Related Works

Jonas Karlsson [6] has proposed a technique for improving TCP performance in the Multihop Wireless Networks. In this technique, the TCP performance is maximized by enhancing the network layer services using two algorithms known as the packet aggregation algorithm and the aggregation aware multipath forwarding algorithm. The packet aggregation algorithm is an algorithm which can be deployed as an IP layer deployment along with already present hardware, else can be deployed with some small alterations in the MAC layer. The aggregation aware forwarding algorithm along with the packet aggregation algorithm enhances the network performance in a multi radio environment. This is beneficial since this algorithm needs lesser exchange of cross layer information.

Ms. Sumedha *et al.*, [7] have proposed an algorithm to Improve Performance over Multihop Wireless Mesh Network. This algorithm estimates the outstanding packets and accordingly decides the slow staart threshold (ssthresh) value which is obtained by calculating the difference between the highest packet transmitted and latest acknowledgement received, and the then by computing the half value of the difference. This difference value is used to handle the regular retransmission timeout which is a result of the retransmission loss. The selection of technique such as Slow Start or congestion Avoidance is based on the ssthresh value. The Slow Start and the Congestion Avoidance of TCP NewReno is selected by the TCP SAC and adapts the Fast Retransmit and Recovery Algorithms.

Maninder Kaur *et al.*, [8] have proposed Snoop Protocol to enhance the TCP performance of Wired and Wireless network. The Snoop protocol is designed to enhance the TCP throughput in both wired as well as the wireless network, since TCP is prones to several issues in these networks. Initially due to the usage of the Snoop protocol, a slight ditch is observed in the TCP performance as a result of the premature retransmissions which happens due to decreased retransmission time out. However the retransmission timeout is set larger to improve the initial performance. The Snoop protocol is implemented at each node along the transmission path and an improvement is observed in the TCP performance.

Changhee Joo *et al.*, [9] have proposed a technique for Synchronizing TCP with Block Acknowledgement over Multi-hop Wireless Networks. Block Acknowledgement technique is used to enhance the TCP performance in the multihop wireless network. Enhancement in the TCP performance is achieved by minimizing the number of acknowledgments used at the manageable risk of reliability. Next the LECN allows the TCP to adjust its window size appropriately. The acknowledgment received after certain delay is included in the block acknowledgment and then similarly the receiver responds to the block of packets through block acknowledgment. The block size is decided upon by the sender node as the block size is basically dependent on the sender's transmission window size. The boundary of the block is described by the sender to the receiver through a single bit block indicator.

May Zin Oo *et al.*, [10] have proposed a Proxy Acknowledgement Mechanism for TCP Variants in Mobile Ad Hoc Networks. The Proxy Acknowledgment (PACK) technique determines the missed out TCP sequence number during its transmission from the source node to the destination node. The PACK technique is applied on each intermediate node to keep a check on the TCP sequences. This technique is used on dynamic network topologies of the wireless network and its variation in performance is analyzed. When the PACK technique is used in the grid topology instead of the PART technique with TCP variants, throughput enhancement is observed along with reduced packet loss of around 60%. When the PACK technique is used in the random topology, a variation in the node speed is seen from 1m/s to 20m/s.

3. Proxy based Intelligent Acknowledgment Technique

3.1. Overview

In this paper, proxy based intelligent acknowledgment technique for multihop networks is proposed. In this technique, proxy nodes are selected among the intermediate nodes along the path from source to destination. The proxy selection is made according to [10]. The Proxy node performs the bandwidth and delay estimation instead of each hop. In addition to this, it also checks the TCP missing sequence numbers. Then it forwards the estimated bandwidth and delay information to the next proxy towards the destination. If it detects some missing packets, then it sends an ACK packet towards the source indicating the missing packets to be re-transmitted. The functions of the receiver node are same as our previous works [2, 14].

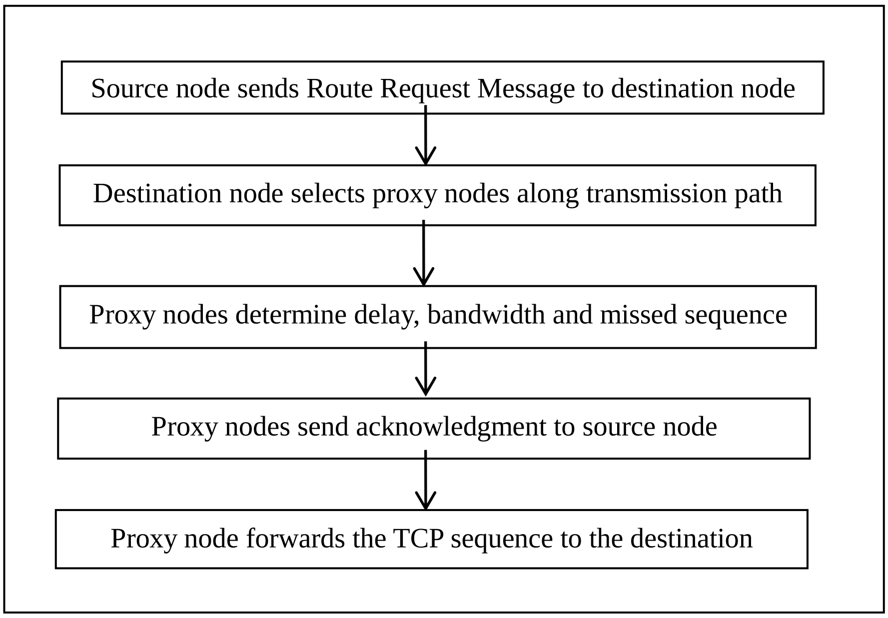


Figure 1. Block Diagram

3.2. Proxy Node Selection

The proxy node is selected to perform the bandwidth and delay calculation. This frees the remaining nodes involved in data transmission from the task of bandwidth and delay estimation. This reduces the overall network delay and saves the network resources to a considerable extent. The selection of the proxy node is performed by the destination node. The proxy node selection process [10] is described in algorithm 1.

Algorithm 1

Notations

S	Source Node
D	Destination Node
RREQ	Route Request
RREP	Route Reply
HC	Hop Count
P	Proxy Node
P_HC	Proxy Hop Count
HC_{Th}	Threshold Hop Count
N_i	Intermediate Node along S to D

---

1. S transmits RREQ to D
2. If D receives RREQ, then
3. D determines the HC between S and D
4. If HC < HC_{Th}, then


```
        D does not select any P
        D sends the RREP which includes the shortest path
Else
        D opts to use the assistance of the P
        D calculates P_HC
        D attaches  P_HC into RREP  and send it toward S.
Do
        If Ni receives RREP, the
              Ni retrieves P_HC from RREP
              If P_HC = HC, then
                     Ni is selected as Proxy
              Else
                     RREP packet is forwarded to  Ni+1
              End if
        End if
        HC =1
        i = i+1
While (RREP reaches S)
End if
End if
```

When S wants to transmit a data packet to the D, it initially sends a RREQ packet to the D. On receiving the RREQ packet, the D determines the HC between the S and D. If HC is less than a threshold value, then D does not select any P and sends the RREP which includes the shortest path. Otherwise, D opts to use the assistance of the P. To select P, the D invokes the proxy calculation function. The proxy calculation function calculates suitable P_HC value. D attaches the P_HC value in the RREP packet and sends it towards the S. Every intermediate node which receives the RREP packet, retrieves the P_HC value and compares it with the HC of the RREP packet. If HC and P_HC values are not equal, then RREP packet is forwarded to the next intermediate node. Otherwise, the intermediate node is selected as the proxy node. RREP packet is again forwarded with the HC reset to 1. The intermediate nodes along the path till S compare the HC and P_HC. If it matches then the intermediate node is set as the next P. Otherwise, RREP is forwarded until it reaches S.

In this way, the proxy nodes are selected based on the appropriate distance between the source node and the destination node. So, in case of any error or packet missing situation, the proxy nodes can send information to the source node.

3.3. Detection of the Missing TCP packets

The proxy nodes perform functions like estimation of the delay involved, bandwidth consumed and detection of the missing TCP packets. Based on the detection, the proxy node notifies the source node of the missing packets and requests it to resend them. This process is described in algorithm 2.

Algorithm 2

Notations

P	Proxy Node
T	Time Delay
t_i	queuing delay while entering link layer
t_o	queuing delay while leaving link layer
t_t	time required for the data packet delivery
t_d	time required for single data transmission
SIFS	Short inter-frame space
t_{ack}	time computed during the reception of ACK packets.
DIFS	Distributed Coordination Function Inter-Frame Space
BW	Bandwidth
pkt_size	data packet size
t	time taken to transmit data packet
rxd_seq_num	received sequence number
expected_seq_num	expected sequence number
present_proxy	current proxy
missed_seq_num	sequence number of the missed TCP sequence
missed_num	number of missed sequence
ACK	Acknowledgment
neg_ACK	negative Acknowledgment

1. Each node along the transmission path records the nearest P ID in its routing table.

2. When a P receives a TCP packet, it retrieves the corresponding sequence number and adds it in the rxd_seq_num field in the routing table.

3. The P estimates the T involved in transmission according to equation 1.

$$T = t_i - t_o + t_t \quad (1)$$

Where $t_t = t_d + SIFS + t_{ack} + DIFS$

4. The BW is estimated according to the equation 2.

$$BW = pkt\_size/t \quad (2)$$

5. If the rxd_seq_num = 1, then it is added in the expected_seq_num field and the P ID is added in the present_proxy field in the routing table.

6. If this P ID in the present_proxy field ≠ current proxy node ID, then the rxd_seq_num is added in the expected_seq_num field in the routing table of the proxy node.

7. Then the corresponding node ID is added in the present_proxy field in the routing table.

8. After the P ID is determined according to the current network situation, then the P analyzes the rxd_seq_num.

9. If the rxd_seq_num > expected_seq_num, then the expected_seq_num is added in the missed_seq_num field in the routing table.

10. Then the number of missing TCP sequences are estimated according to equation 3.

$$missed\_num = rxd\_seq\_num - expected\_seq\_num \quad (3)$$

11. Then the expected_seq_num field is added with the rxd_seq_num.

12. If the rxd_seq_num < expected_seq_num, then the expected_seq_num field is update with the value obtained from equation 4.

$$expected\_seq\_num = \max(expected\_seq\_num, rxd\_seq\_num) \quad (4)$$

13. If the rxd_seq_num = expected_seq_num, then the expected_seq_num field is updated with value obtained from equation 5.

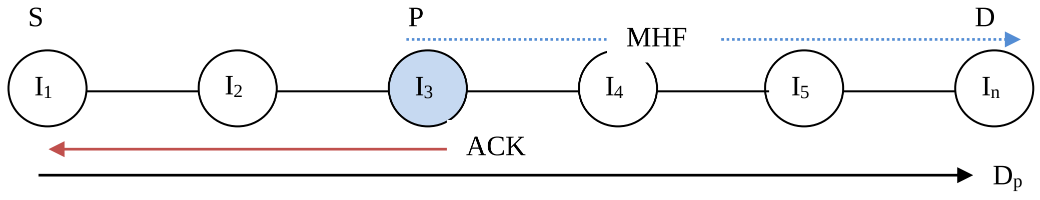
$$expected\_seq\_num = expected\_seq\_num + 1 \quad (5)$$

14. For each rxd_seq_num, the P sends an ACK to the S and forwards the sequence to the destination through intermediate nodes.

15. For each missed_seq_num, the P sends a neg_ACK packet to the source which includes the missed_seq_num, missed_num, T and BW.

16. On receiving this message the S retransmits the missed TCP sequence.

Thus, the proxy node analyzes each received TCP sequence and determines the missed sequence. Since this information is instantly sent to the transmitting node, quick re transmission is possible. This minimizes the delay in network operation and maximizes the network efficiency.



S Source D Destination P Proxy

Figure 2. Proxy based Intelligent Acknowledgement Technique

Figure 2 shows the Proxy based Intelligent Acknowledgement technique. Here the nodes $I_1, I_2, \ldots I_n$ forms the multihop network. Let the source S send a TCP data packet to the destination D. The proxy node I_3 estimates the bandwidth and delay and transmits the updated MAC header field (MHF) to D. P also estimates the sequence numbers and transmits ACK packet to S.

4. Simulation Results

4.1. Simulation Parameters

The proposed Proxy based TCP Intelligent Acknowledgment Technique (PTCP-IA) is simulated using NS-2 [15]. The area size is 1500 meter x 1500 meter square region. The simulation topology is shown in Figure-3. In this figure, nodes 6, 9 and 12 are deployed as proxy nodes and nodes 0 and 15 act as the source and destination nodes, respectively. The proposed PTCP-IA is compared with the traditional TCP New Reno protocol. The performance is evaluated based on end-to-end delay, packet delivery ratio and received throughput.

The simulation settings and parameters are summarized in Table 1.

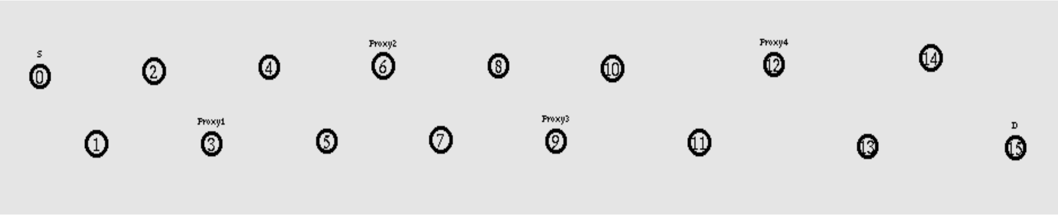


Figure 3. Simulation Topology

Table 1. Simulation Settings

Number of Nodes	16
Area Size	1500 X 1500m
MAC Protocol	802.11
Simulation Time	50 sec
Packet Size	250,500,750,1000and 1250 bytes
Propagation	TwoRayGround
Antenna	OmniAntenna
Time	10,15,20,25 and 30

4.2. Results & Analysis

The simulation results are presented in the next section.

A. Based on Packet Size

The TCP packet size is varied as 250,500,750,1000 and 1250 bytes with simulation time interval as 25 seconds.

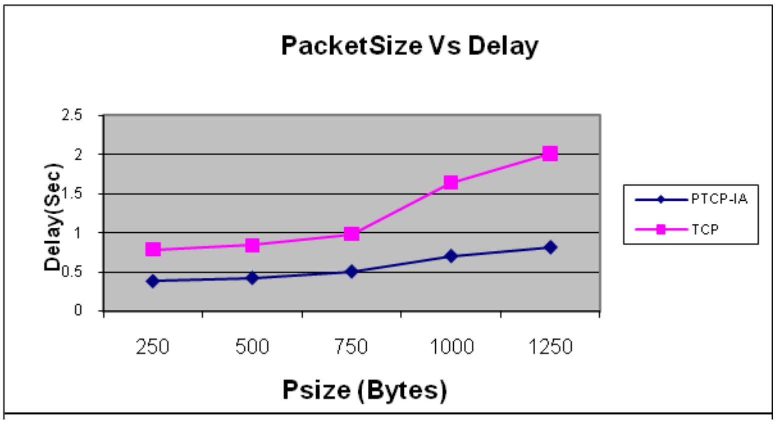


Figure 4. Packet Size Vs Delay

Figure 4 shows the delay occurred for PTCP-IA and TCP when the packet size is varied. The increase in packet size results in increase in delay. As seen from the figure, the delay of PTCP-IA increases from 0.38 to 0.81 seconds and the delay of TCP increases from 0.79 to 2.01 seconds. PTCP-IA has 53.13% lesser delay than TCP technique, since it uses proxy nodes for acknowledgements.

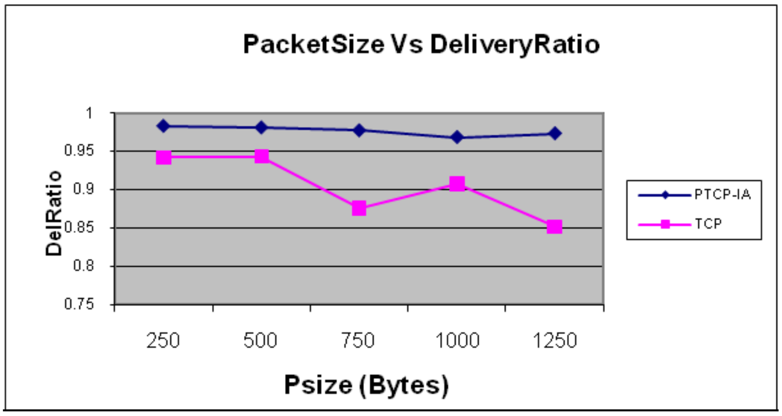


Figure 5. Packet Size Vs Delivery Ratio

Figure 5 shows the packet delivery ratio obtained for PTCP-IA and TCP when the packet size is varied. The increase in packet size results in decrease in delivery ratio. As seen from the figure, the delivery ratio of PTCP-IA decreases from 0.98 to 0.97 and the delivery ratio of TCP decreases from 0.94 to 0.85. However, PTCP-IA has 7.4% higher delivery ratio than TCP, since proxy nodes are applied for acknowledgements along with bandwidth adjustments.

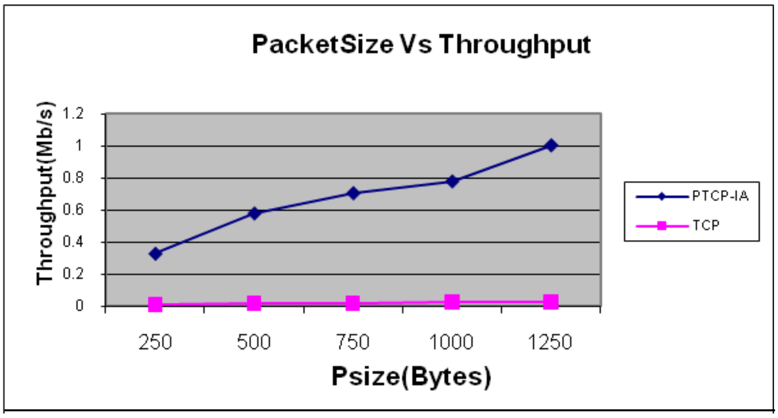


Figure 6. Packet Size Vs Throughput

Figure 6 shows the throughput obtained for PTCP-IA and TCP techniques when the packet size is varied. The increase in packet size results in increase in throughput. As seen from the figure, the throughput of PTCP-IA increases from 0.32 to 1.00 Mb/sec and the throughput of TCP increases from 0.01 to 0.02 Mb/sec. But PTCP-IA has 97.07% higher throughput than TCP technique, since proxy nodes are applied for acknowledgements along with bandwidth and delay adjustments.

B. Based on Time Intervals

The simulation time interval is varied from 10 to 30 seconds with packet size 500 bytes.

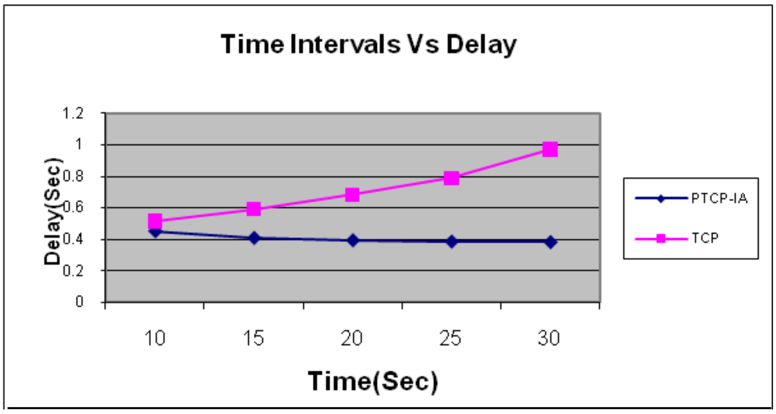


Figure 7. Time Vs Delay

Figure 7 shows the delay occurred for PTCP-IA and TCP techniques when the time interval is varied. The increase in time results in increase in delay of TCP. As seen from the figure, the delay of TCP increases from 0.51 to 0.97 seconds. For PTCP-IA, the delay remains almost constant around 0.45 seconds. PTCP-IA has 39.44% lesser delay than TCP, since it uses proxy nodes for acknowledgements.

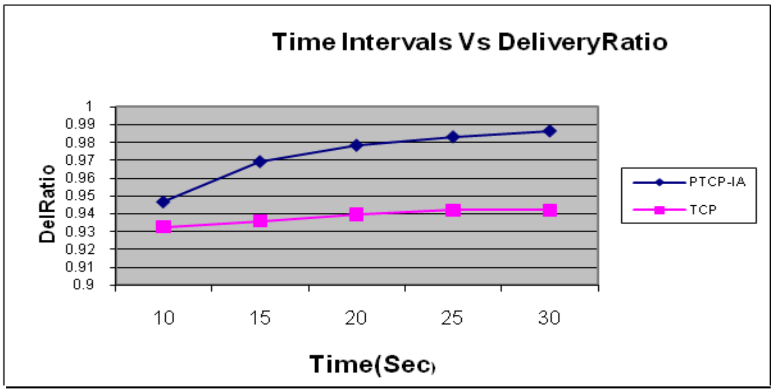


Figure 8. Time Vs Delivery Ratio

Figure 8 shows the packet delivery ratio obtained for PTCP-IA and TCP techniques when the time interval is varied. The increase in time results in increase in delivery ratio. As seen from the figure, the delivery ratio of PTCP-IA increases from 0.94 to 0.98 and the delivery ratio of TCP increases from 0.93 to 0.94. However PTCP-IA has 4% higher delivery ratio than TCP since proxy nodes are applied for acknowledgements along with bandwidth adjustments.

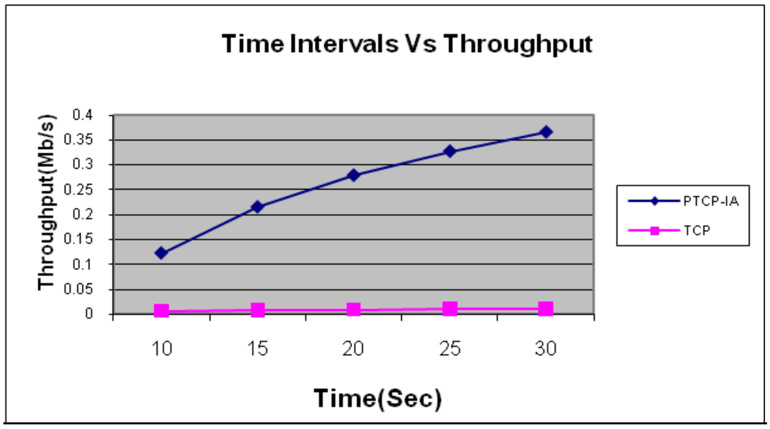


Figure 9. Time Vs Throughput

Figure 9 shows the throughput obtained for PTCP-IA and TCP techniques when the time interval is varied. The increase in time interval results in increase in throughput. As seen from the figure, the throughput of PTCP-IA increases from 0.12 to 0.36 Mb/sec and the throughput of TCP increases from 0.006 to 0.01 Mb/sec. But PTCP-IA has 96.22% higher throughput than TCP technique, since proxy nodes are applied for acknowledgements along with bandwidth and delay adjustments.

5. Conclusion

In this paper, a Proxy based Intelligent Acknowledgment Technique is developed for multi hop networks. In this technique, when the source node wants to transmit the data, it sends a route request message to the destination node. On receiving this request, the destination node selects proxy nodes along the transmission path. Then the source node starts transmitting the TCP packet sequences. On receiving each sequence, the proxy node determines the delay, bandwidth and missed TCP sequences. For each received sequence, the proxy node sends an acknowledgment. If any sequence is missed out, then the proxy node sends negative acknowledgment to the source requesting retransmission. Simulation results show that, PTCP-IA outperforms TCP in terms of delay by 53%, delivery ratio by 7.4%, throughput by 97%, when the number packet size is varied from 250 to 1250 Bytes. Similarly, it outperforms TCP in terms of delay by 39%, delivery ratio by 3.5% and throughput by 96%, when the simulation time is varied from 10 to 30 seconds.

Acknowledgments

I would like to thank Professor Arti Noor at CDAC Noida, for contributing the idea of Proxy based Intelligent Acknowledgment Technique in Multihop Wireless Network. Without her kind support and assistance this work would not be possible.

References

[1] T. Braun, A. Kassler, M. Kihl., V. Rakocevic, V. Siris and G. Heijenk, Technical Report., “Multi-Hop Wireless Networks”, Chapter-5.

[2] V. Chawla., “H*H: Hop by Hop Analysis of TCP-IA for Multi-Hop Wireless Networks”, International Journal of Computer Technology & Applications, vol. 5, no. 3, **(2014)**, pp. 1276-1281.

[3] K. Jain, J. Padhye, V. Padmanabhan and L. Qiu, “Impact of Interference on Multi-hop Wireless Network Performance”, MobiCom, San Diego, California, USA, ACM 1-58113-753-2/03/0009, **(2003)**.

[4] S. M. ElRakabawy, C. Lindemann and M. K. Vernon, “Improving TCP Performance for Multihop Wireless Networks”, Proceedings of the 2005 International Conference on Dependable Systems and Networks (DSN’05), 0-7695-2282-3/05, **(2005)**.

[5] Z. Fu, H. Luo, P. Zerfos, S. Lu, L. Zhang and M. Gerla, “The Impact of Multihop Wireless Channel on TCP Performance”, IEEE Transactions on Mobile Computing, **(2005)**.

[6] J. Karlsson, "Improving TCP Performance in Wireless Multi-hop Networks", Karlstad University Studies 2011:5, ISSN 1403-8099, ISBN 978-91-7063-337-9, Print: Universitetstryckeriet, **(2010)**.
[7] M. Sumedha, R. Chokhandre and Ms. U. Shrawankar, "An Algorithm to Improve Performance over Multihop Wireless Mesh Network", Journal of Computing, ISSN 2151-9617, vol. 3, no. 5, **(2011)**.
[8] M. Kaur and P. Singh, "Enhancement of TCP Performance with the Help of Snoop Protocol in Wired CUM Wireless Network", International Journal of Distributed and Parallel Systems (IJDPS), vol. 3, no. 6, **(2012)**.
[9] C. Joo, S. Bahk and H. Kim, "Synchronizing TCP with Block Acknowledgement over Multi-hop Wireless Networks", I. Chong and K. Kawahara (Eds.): ICOIN 2006, LNCS 3961, Springer-Verlag, **(2006)**, pp. 267-275.
[10] M. Zin Oo., M. Othman and T. O'Farrell, "A Proxy Acknowledgement Mechanism for TCP Variants in Mobile Ad Hoc Networks", **(2014)**, pp. 1229-2370.
[11] R. Verma, A. Prakash, R. Tripathi and N. Tyagi, "Improving Throughput of Starved TCP Flow by Sidestepping Bottleneck Nodes Using Concurrent Transmission", International Journal of Mobile Computing and Multimedia Communications, vol. 2, no. 1, **(2010)**, pp. 68-83.
[12] J. Wang, J. Wen, J. Zhang and Y. Han, "A demonstration of a new TCP congestion control algorithm over LTE and other challenging networks", MobiCom, **(2010)**.
[13] H. Khalife, S. Mahfoudh and N. Malouch, "A Hop-by-hop Node Failure Control Protocol for Wireless Ad Hoc Networks: Towards MAC layer Control", In the proceedings of the IEEE pimrc conference, **(2007)**.
[14] V. Chawla, "Performance Analysis of TCP-IA for Multi-Hop Wireless Networks", International Journal of Emerging Trends in Engineering and Development, vol. 2, no. 3, **(2013)**.
[15] Network Simulator: http:///www.isi.edu/ns/nam.

