ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 3, Issue 8, August 2015**

# Survey of TCP Adaption in Mobile Ad-Hoc Networks

Kanchana .A[1], P. Gayathri[*2]

[1] Assistant Professor, Department of Information Technology, Jerusalem College of Engineering, Chennai, Tamil Nadu, India

[2]Assistant Professor, Department of Information Technology, Bharath University, Chennai, Tamil Nadu, India

[*] Corresponding Author

**ABSTRACT***:* This paper presents an overview of how TCP can be adapted to Ad hoc Networks. TCP was originally designed to provide reliable end to end delivery. To adapt TCP over the mobile Adhoc Networks, it is necessary to classify and identify the losses. Network congestion is not the only reason for loss in Ad hoc Networks.TCP can be affected by the mobility of the nodes, route failures, wireless channel contention, unfairness. This paper compares and examines the main approaches which would help in adapting TCP to mobile Ad hoc networks environment.

**KEYWORDS:** Mobile networks, Adhoc Networks, TCP survey, packet loss, route failure

## I. INTRODUCTION

Ad hoc networks are complex distributed systems that consist of wireless mobile or static nodes that can freely and dynamically self-organize. In this way they form arbitrary and temporary, “ad hoc” network topologies, allowing devices to seamlessly interconnect in areas with no pre-existing infrastructure.

The Transmission Control Protocol (TCP) was designed to provide reliable end-to-end delivery of data over unreliable networks. In practice, most TCP deployments have been carefully designed in the context of wired networks. Ignoring the properties of wireless ad hoc networks can lead to TCP implementations with poor performance. In order to adapt TCP to the ad hoc environment, improvements have been proposed in the literature to help TCP to differentiate between the different types of losses.

## II. PROBLEMS OF TCP IN AD HOC NETWORKS:

TCP is a connection-oriented transport layer protocol that provides reliable, in-order delivery of data to the TCP receiver. Usage of TCP without any modification to suit mobile ad hoc networks, results in a serious drop in connection’s throughput. There are several reasons for such a drastic drop in TCP throughput and below sections examines these reasons in brief.

*Effect of a High BER:* Bit errors causes the packets to get corrupted which result in lost TCP data segments or acknowledgment. When acknowledgment do not arrive at the TCP sender within a short amount of time [the retransmit timeout (RTO)],the sender retransmits the segment, exponentially backs off its retransmit timer for the next retransmission, reduces its congestion control window threshold, and *closes its congestion window to one segment.* Repeated errors will ensure that the congestion window at the sender remains small resulting in low throughput [1], [3]. It is important to note that error correction may be used to combat high BER but it will waste valuable wireless bandwidth when correction is not necessary.

*Effect of Route Recomputations:* When an old route is no longer available, the network layer at the sender attempts to find a new route to the destination [in dynamic source routing (DSR) [8] this is done via route discovery messages

DOI: 10.15680/IJIRCCE.2015. 0308210

while in destination-sequenced distance-vectoring (DSDV) [11] table exchanges are triggered that eventually result in a new route being found]. It is possible that discovering a new route may take significantly longer than the RTO at the sender. As a result, the TCP sender times out, retransmits a packet, and invokes congestion control. Thus, when a new route is discovered, the throughput will continue to be small for some time because TCP at the sender grows its congestion window using the slow start and congestion avoidance algorithm. This is clearly undesirable behavior because the TCP connection will be very inefficient. If we imagine a network in which route computations are done frequently (due to high node mobility), the TCP connection will never get an opportunity to transmit at the maximum negotiated rate (i.e., the congestion window will always be significantly smaller than the advertised window size from the receiver).

*Effect of Network Partitions:* It is likely that the *ad hoc* network may periodically get partitioned for several seconds at a time. If the sender and the receiver of a TCP connection lie in different partitions, all the sender's packets get dropped by the network resulting in the sender invoking congestion control. If the partition lasts for a significant amount of time (say, several times longer than the RTO), the situation gets even worse be-cause of serial *timeouts*. A serial timeout is a condition wherein multiple consecutive retransmissions of the same segment are transmitted to the receiver while it is disconnected from the sender. All these retransmissions are, thus, lost.

The following are the major problems which reduce the performance of TCP

- TCP is unable to distinguish between losses due to route failures and losses due to network congestion.
- TCP suffers from frequent route failures.
- The contention on the wireless channel.
- TCP unfairness.
- TCP performance in such networks suffers from significant throughput degradation and very high interactive delays

## III. RELATED WORK

### *TCP PERFORMANCE OVER MANETS:*

Monks *et al.* [2], investigated the impact of mobility on TCP throughput in MANETs. In their simulation scenarios, nodes move according to the random way-point model with 0s pause time. The speed of node was uniformly distributed for some mean speed *v*. DSR was used at the routing layer and it was found that when the mean speed increases from 2m/s to 10m/s the throughput drops sharply. But, when it increases from 10m/s to 30m/s the throughput drops slightly. It was also observed that, for a given mean speed, certain mobility patterns achieve throughput close to 0, although the other mobility patterns are able to achieve high throughput. By analyzing the simulations trace of patterns of low throughput, they found that the TCP sender's routing protocol is unable to quickly recognize and purge stale routes from its cache, which results in repeated routing failures and TCP retransmission timeouts. For patterns of high throughput they found that most of time the TCP sender and receiver are close to each other. By examining the mobility patterns, the authors observe that as the sender and receiver move closer to each other, DSR can maintain a valid route. This is done by shortening the existing route before a routing failure occurs. However, as the sender and receiver move away from each other, DSR waits until a failure occurs to lengthen a route.

To prevent TCP invocation of congestion control that deteriorates TCP throughput in case of losses induced by mobility, Monks *et al.* suggest the usage of explicit link failure notification (ELFN) technique. "TCP treats losses induced by route failures as signs of network congestion", Anantharaman et al. identify a set of factors that contribute also to the degradation of TCP throughput in the presence of mobility. These factors are: MAC failure detection and route computation latencies. The MAC failure detection latency is defined as the amount of time spent before the MAC concludes a link failure. They found that in the case of the IEEE 802.11 protocol, when the load is light (one TCP connection) this latency is small and independent of the speed of the nodes. However in case of high load, the value of this latency is magnified and becomes a function of the nodes' speed. The route computation latency is defined as the time taken to recompute the route after a link failure. They found that, as for MAC failure detection latency, the route computation latency increases with the load and becomes a function of the nodes' speed in the high load case. Also, the authors identify another problem, called MAC packet arrival that is related to routing protocols. In fact, when a link failure is detected, the link failure is sent to the routing agent of the packet that triggered the detection. If other sources are using the same link in the path to their destinations, the node that detects the link failure has to wait till it receives a packet from these sources before they are informed of the link failure. This also contributes to the delay after which a source realizes that a path is broken.

*TCP PERFORMANCE OVER SANETS:*

In [3][4][5][6] the authors report simulation results on TCP throughput in a static linear multi-hop chain, where IEEE 802.11 protocol is used. In Figure 4, we display a multi-hop chain of *N* nodes. It is expected that, as the number of hops increases, the spatial reuse will also increase. However, simulation results indicate that TCP throughput decreases "rapidly" up to a point as the number of hops increases. It is argued that this decrease is due to the hidden terminals problem, which increases frames collisions. After a repeated3 transmission failure MAC layer will react by two actions. First, the MAC will drop the head-of-line frame destinated to the next hop, we note that this type of drops is known also as drops due to contention on wireless channel. Second, the MAC will notify the upper layer about a link failure. When the routing protocol of a source node detects a routing failure, it will initiate a route re-establishment process. In general the route re-establishment duration is greater than the retransmission timer of the TCP agent; hence the TCP agent will enter the backoff procedure and will set its congestion window to 1. Also, as TCP sender's does not have indications on the route re-establishment event, TCP will suffer from a long idle time. During this time, the network may be connected again, but TCP is still in the back-off state.

In [8], Xu and Saadawi study the performance of TCP Tahoe, Reno, New Reno, Sack and Vegas4 over the multi-hop chain topology shown in Figure 4, in the case where the IEEE 802.11 protocol is used. It is shown that TCP Vegas delivers the better performance and does not suffer from instability. By tuning the sender TCP's maximum window size (advertised window) to approximately four packets, all TCP versions perform similarly.

Furthermore, the authors investigate the performance of these TCP versions using the delayed-ACK option, as defined in RFC 1122. According to the mentioned RFC, the TCP sink will send one TCP ACK packet for every two TCP packets received. This option will reduce the contention on the wireless channel, because the data and ACK packets share the same wireless channel. Simulating the multi-hop chain with the delayed-ACK option, they report that an improvement of 15% to 32%.

## IV. PROPOSALS FOR TCP ADAPTATION

### *A.PROPOSALS TO REDUCE ROUTE FAILURES*

The proposals that address the problem of frequent route failures in MANETs can be classified as follows to three categories: TCP and network cross layer proposals, network and physical cross layer proposals, and network layer proposals.[1]

*1)* **Split TCP:** TCP connections that have large number of hops suffer from frequent route failures due to mobility. To improve the throughput of these connections and to resolve the unfairness problem, the Split TCP scheme was introduced to split long TCP connections into shorter localized segments [10] – see Figure 6. The interfacing node between two localized segments is called a proxy. The routing agent decides if its node has the role of proxy according to the inter-proxy distance parameter[2]. The proxy intercepts TCP packets, buffers them, and acknowledges their receipt to the source (or previous proxy) by sending a local acknowledgment (LACK). Also, a proxy is responsible 19 for delivering the packets, at an appropriate rate, to the next local segment. Upon the receipt of a LACK (from the next proxy or from the final destination), a proxy will purge the packet from its buffer. To ensure the source to destination reliability, an ACK is sent by the destination to the source similarly to the standard TCP. In fact, this scheme splits also the transport layer functionalities into those end-to-end reliability and congestion control. This is done by using two transmission windows at the source which are the congestion window and the end-toend window. The congestion window is a sub-window of the end-to-end window.[3] While the congestion window changes in accordance with the rate of arrival of LACKs from the next proxy, the end-to-end window will change in accordance with the rate of arrival of the end-to-end ACKs from the destination. At each proxy, there would be a congestion window that would govern the rate of sending between proxies.[4]

*2) **Detect using signal strength based link management**:*

In [8] authors propose two mechanisms for alleviating the effects of mobility on TCP performance. We call these the Proactive and the Reactive Link Management (LM) schemes. These schemes are implemented at the MAC layer. We also provide a modification of AODV at the network layer that can exploit the presence of the link management schemes. Proactive LM tries to predict link breakage, whereas Reactive LM temporarily keeps a broken link alive with higher transmission power to salvage packets in transit. In this method they proactively determine the signal strength and if its below the threshold they reroute the packets thus avoiding the link failures.[5]

*3) **MultiPath Routing:***

In [9] the authors investigate an alternative way of utilizing multipath routing. That is, to let TCP only use one path at a time and keep the other paths as backup routes, which we refer as backup path multipath routing. Backup path multipath routing still maintains several paths from a source to a destination. However, it only uses one path at any time. When current path breaks, it can quickly switch to other alternative paths.[6]

*B.PROPOSALS TO CLASSIFY THE PACKET LOSS:*

1) **TCP-f-** It is a feedback-based approach [10] to handle route failures in MANETs. A separate notification packet (RFN) is sent to indicate the route failure. On receiving the RFN, the source goes into a snooze state.

2) **ELFN-based Technique-** This interaction aims to inform the TCP agent about route failures when they occur. The authors use an ELFN [11] message, which is piggybacked onto the route failure message sent by the routing protocol to the sender.

3) **ATCP[14]-** To detect packet losses due to channel errors, ATCP monitors the received ACKs. When ATCP sees that three duplicate ACKs have been received, it does not forward the third duplicate ACK but puts TCP in the persist state and quickly retransmits the lost packet from TCP's buffer.[12]

4) **TCP-BuS-** [13]It uses network feedback to detect route failure events and to take convenient action in response to these events.[7] The novel scheme in this proposal is the introduction of *buffering capability* in mobile nodes

*C.PROPOSALS TO IMPROVE TCP FAIRNESS*

***1) Enhanced RED:*** Significant TCP unfairness in ad hoc wireless networks has been reported during the past several years. This unfairness results from the nature of the shared wireless medium and location dependency.[8] If we view a node and its interfering nodes to form a "neighborhood", the aggregate of local queues at these nodes represents the distributed queue for this neighborhood. However, this queue is not a FIFO queue.[9] Flows sharing the queue have different, dynamically changing priorities determined by the topology and traffic patterns. Thus, they get different feedback in terms of packet loss rate and packet delay when congestion occurs. In wired networks, the Randomly Early Detection (RED) scheme was found to improve TCP fairness.[10] In this paper, we show that the RED scheme does not work when running on individual queues in wireless nodes. We then propose a Neighborhood RED [50](NRED) scheme, which extends the RED concept to the distributed neighborhood queue. Simulation studies confirm that the NRED scheme can improve TCP unfairness substantially in ad hoc networks.

Moreover, the NRED scheme acts at the network level, without MAC protocol modifications. This considerably simplifies its deployment.[11]

***2) Adaptive_Pause :*** In wireless ad hoc networks, fair allocation of bandwidth among different TCP flows is one of the critical problems that affect the performance of the entire system. This paper proposes a fairness mechanism - Adaptive_Pause.[12] Adaptive_Pause is a simple and distributed scheme that only needs little communication and processing overhead. Each node monitors the occupation of the channel due to its emissions and dynamically determines whether it should pause a time interval in order to avoid channel capture. Comparing to other passive schemes in which a node limits its transmission when it receives congestion indication from its neighbors, the proposed scheme is more effective and requires less overhead. Simulation result validated the analysis result and gave the optimal parameter setting.[13] Both analytic and simulation results show that Adaptive_Pause scheme can improve the TCP fairness

***3) Cross layer approach to Enhance TCP Fairness:***[17]
The basic idea of this approach is to exploit the advertised window field (*adw*) of TCP segments to limit the transmission rate of the TCP sender1. As well known [18], the TCP sender cannot have a number of outstanding segments larger than the *adw* value advertised by its own receiver. Currently, the TCP receiver stamps into the *adw* field the space available into its receiving buffer, in order to avoid saturation by a fast connection.[14] We propose to extend the use of *adw* in order to allow the receiver to limit the transmission rate of 1The algorithm has been conceived for unidirectional TCP flows, even if it could be easily extended to the bidirectional case with only some slight modifications. the TCP sender also when the path used by the connection exhibits a high frame collision probability.[15] In particular, in the proposed cross-layer approach measurements about frame collision probability along the path are collected at the MAC layer and are communicated to the TCP receiver in order to properly set *adw*. To this aim, we assume the presence of a specific field, known as *nonCollisionProb* field, in the MAC Protocol Data Unit (MPDU).[16] This field is set equal to the non-collision probability *pnc* at link level. In particular, *pnc* is set equal to one in frames transmitted at the first hop of a TCP connection. At intermediate nodes, instead, *pnc* is set by taking into account the collision probability *pc* estimated by the wireless
nodes from the ratio between the number of retransmitted frames and the total number of transmitted frames. After a frame reception, each node, before the forwarding, sets the *pnc* value

### *D.PROPOSALS TO REDUCE WIRELESS CHANNEL CONTENTION:*

***1) Rate control based on channel utilization and contention:*** In ad hoc networks, both contention and congestion can severely affect the performance of TCP. In this paper the author shows that the over-injection of conventional TCP window mechanism results in severe contentions, and medium contentions cause network congestion.[17] Furthermore, introducing two metrics, channel utilization (CU) and contention ratio (CR), we characterize the network status. Then, based on these two metrics, we propose a new TCP transmission rate control l mechanism based on Channel utilization and Contention ratio (TCPCC). In this mechanism, each node collects the information about the network busy status and determines the CU and CR accordingly. The CU and CR values fed back through ACK are ultimately determined by the bottleneck node along the flow. The TCP sender controls its transmission rate based on the feedback information. [18]

**2) Dynamic delayed Ack:** This approach [19] aims to reduce the contention on wireless channel, by decreasing the number of TCP ACKs transmitted by the sink. It is a modification of the delayed ACK option (RFC 1122) that has a fixed coefficient $d = 2$. In fact, $d$ represents the number of TCP packets that the TCP sink should receive before it acknowledges these packets. In this approach, the value of $d$ is not fixed and it varies dynamically with the sequence number of the TCP packet. For this reason, the authors define three thresholds $l1$, $l2$, and $l3$ such that $d = 1$ for packets with sequence number $N$ smaller than $l1$, $d = 2$ for packets with $l1 \cdot N \cdot l2$, $d = 3$ for $l2 \cdot N \cdot l3$ and $d = 4$ for $l3 \cdot N$. In their simulations, they study the packet loss rate, throughput, and session delay of TCP New Reno, in the case of short and persistent TCP sessions on a static multihop chain. They show that their proposal, with $l1 = 2$, $l2 = 5$, and $l3 = 9$, outperforms the standard TCP as well as the delayed ACK option for a fixed coefficient $d = 2; 3; 4$. They suggest that better performance could be obtained by making $d$ a function of the sender's congestion window instead of a function of the sequence number.[19]

**3) COPAS:** COntention-based PAth Selection proposal [19] addresses the TCP performance drop problem due to the contention on the wireless channel.[20] It implements two techniques: the first one is disjoint forward and reverse routes, which consists of selecting disjoint routes for TCP data and TCP ACK packets. The second one is dynamic Contention-balancing, which consists of dynamically update disjoint routes.[21] Once the contention of a route exceeds a certain threshold, called backoff threshold, a new and less contented route is selected to replace the high contended route. In this proposal, the contention on wireless channel is measured as a function of the number of times that a node has backed off during each interval of time[22]. Also at any time a route is broken, in addition to initiating a route re-

establishment procedure, COPAS redirects TCP packets using the second alternate route. Comparing COPAS and DSR, the authors found that COPAS outperforms DSR in term of TCP throughput and routing overheads.[23]
The improvement of TCP throughput is up to 90%. However, the use of COPAS, as reported by the authors, is limited to static networks or networks with low mobility. Because, as nodes move faster using a disjoint forward and reverse routes increases the probability of ro[24]. The modified AODV allows the forwarding of packets in transit on a route that is going down while simultaneously initiating a search for a new route failures experienced by TCP connections. So, this may induce more routing overheads and more packet losses.[25]

## V. CONCLUSION

Four proposals are used to address the route failures. The main cause of route failures is node mobility. Split TCP is based on splitting long TCP connections, in term of hops, into short segments to decrease the number of routing failures. However, this generates more overheads. In Preemptive routing and Signal strength based link management, the problem is addressed by predicting link failures and initiating a route reconstruction before the current route breaks. Signal strength based link management proposal uses a more robust approach to predict failures than Preemptive routing. Backup routing improves the TCP path availability by storing an alternative path. This path is used when a routing failure is detected. Backup routing reports an improvement in TCP throughput of up to 30% with a reduction in routing overheads. But, further evaluations are needed especially for the route selection criteria.

In proposals to classify the packet loss, four proposal like TCP-f, ELFN-based Technique, ATCP, TCP-bus on an explicit notification from the network layer to detect the route failures. But, they differ in how to detect route re-establishments

In Proposals to improve TCP fairness, *Enhanced RED*, Adaptive Pause and a Cross layer approach to Enhance TCP Fairness are proposed. In Non work-conserving scheduling proposal, this is done through penalizing greedy nodes of high output rate, by increasing their queuing delays at the link layer. However, in Enhanced RED it is up to TCP to regulate its transmission rate when it senses packets drop. Enhanced RED is better than Non work-conserving as it does not cause degradation in total throughput.

In Proposals to reduce wireless channel contention, Rate control based on channel utilization and contention, Dynamic delayed Ack, COPAS are proposed Dynamic delayed ACK is a simple approach that aims to reduce the contention on wireless channel by decreasing the number of TCP ACKs transmitted by the sink. However, COPAS attacks this problem by using disjoint forward and reverse routes, and dynamic update of disjoint routes basing one contention level. COPAS reports an improvement in TCP throughput up to 90%.

## REFERENCES

1. Jian Liu and Suresh Singh" ATCP: TCP for Mobile Ad Hoc Networks " in IEEE Journal on selected areas in communications.
2. Gavin Holland and Nitin Vaidya" Analysis of TCP performance over mobile ad hoc networks,"
3. Krishnamoorthy P., Jayalakshmi T., "Preparation, characterization and synthesis of silver nanoparticles by using phyllanthusniruri for the antimicrobial activity and cytotoxic effects", Journal of Chemical and Pharmaceutical Research, ISSN : 0975 – 7384, 4(11) (2012) pp.4783-4794.
4. E. Altman and T. Jim´enez, "Novel delayed ACK techniques for improving TCP performance in multihop wireless networks," inProc.of the Personal Wireless Communications, Venice, Italy, Sep. 2003, pp. 237–253.
5. Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on TCP throughput and loss'," in Proc. of IEEE INFOCOM, San Francisco, USA, Apr. 2003.
6. Madhubala V., Subhashree A.R., Shanthi B., "Serum carbohydrate deficient transferrin as a sensitive marker in diagnosing alcohol abuse: A case - Control study", Journal of Clinical and Diagnostic Research, ISSN : 0973 - 709X, 7(2) (2013) pp.197-200.
7. S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks?," IEEE Communications Magazine, vol. 39, no. 6, pp. 130–137, Jun. 2001.
8. M. Gerla, K. Tang, and R. Bagrodia, "TCP performance in wireless multi-hop networks," in Proc. of the IEEE WMCSA, New Orleans,LA, USA, 1999
9. Khanaa V., Thooyamani K.P., Saravanan T., "Simulation of an all optical full adder using optical switch", Indian Journal of Science and Technology, ISSN : 0974-6846, 6(S6)(2013) pp.4733-4736.
10. Mohammad Amin Kheirandish Fard *, Kamalrulnizam Abu Bakar*,Sasan Karamizadeh**, Roozbeh Hojabri Foladizadeh** ""Improve TCP Performance over Mobile Ad Hoc Network by Retransmission Timeout Adjustment".
11. Improving TCP Performance in Ad Hoc Networks using Signal Strength based Link Management Fabius Klemm , Zhenqiang Ye_ , Srikanth V. Krishnamurthy_ , Satish K. Tripathi_

12. Nagarajan C., Madheswaran M., "Stability analysis of series parallel resonant converter with fuzzy logic controller using state space techniques", Electric Power Components and Systems, ISSN : 1532-5008, 39(8) (2011) pp.780-793.
13. H. Lim, K. Xu, and M. Gerla, “TCP performance over multipath routing in mobile ad hoc networks,” in Proc. of IEEE ICC, Anchorage,Alaska, May 2003.
14. K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, “A feedback based scheme for improving TCP performance in Ad-Hoc wireless networks,” in Proc. of the International Conference on Distributed Computing Systems (ICDCS’98), Amsterdam, Netherlands,May 1998.
15. J. Monks, P. Sinha, and V. Bharghavan, “Limitations of TCP-ELFN for ad hoc networks,” in Proc. of Mobile and Multimedia Communications, Tokyo, Japan, Oct. 2000.
16. K. Ramakrishnan, S. Floyd, and D. Black, “The addition of explicit congestion notification (ECN) to IP,” RFC 3168, Category :Standards Track, Sep. 2001.
17. D. Kim, C. Toh, and Y. Choi, “TCP-BuS: Improving TCP performance in wireless ad hoc networks,” Journal of Communications and Networks, vol. 3, no. 2, pp. 175–186, Jun. 2001.
18. J. Liu and S. Singh, “ATCP: TCP for mobile ad hoc networks,” IEEE JSAC, vol. 19, no. 7, pp. 1300–1315, Jul. 2001
19. K. Xu, M. Gerla, L. Qi, and Y. Shu, “Enhancing TCP fairness in ad hoc wireless networks using neighborhood red,” in Proc. of ACM MOBICOM, San Diego, CA, USA, Sep. 2003, pp. 16–28.
20. Yantai Shu ; Sanadidi, M. ; Gerla, M “ A Method for Improving the TCP Fairness in Wireless Ad Hoc Networks” in Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference
21. Camarda, P. ; Grieco, L.A. ; Mastrocristino, T. ; Tesoriere “ A Cross-layer Approach to Enhance TCP Fairness in Wireless Ad-hoc Networks “Wireless Communication Systems, 2005. 2nd International Symposium .
22. Bhat V., "A close-up on obturators using magnets: Part I - Magnets in dentistry", Journal of Indian Prosthodontist Society, ISSN : 0972-4052 , 5(3) (2005) pp.114-118.
23. Nana Li ; Wenbo Zhu ; Dan Sung “TCP transmission rate control mechanism based on channel utilization and contention ratio in AD hoc networks “ in Communications Letters, IEEE.
24. E. Altman and T. Jim´enez, “Novel delayed ACK techniques for improving TCP performance in multihop wireless networks,” in Proc.of the Personal Wireless Communications, Venice, Italy, Sep. 2003, pp. 237–253.
25.M.Sundararajan & R.Pugazhanthi,” Human finger print recognition based biometric security using wavelet analysis”, Publication of International Journal of Artificial Intelligent and Computational Research, Vol.2. No.2. pp.97-100(July-Dec 2010).
27.M.Sundararajan & E.Kanniga,” Modeling and Characterization of DCO using Pass Transistor”, proceeding of Springer – Lecturer Notes in Electrical Engineering-2011 Vol. 86, pp. 451-457(2011). ISSN 1876-1100.( Ref. Jor- Anne-II)
28.M.Sundararajan & C.Lakshmi, “Wavelet based finger print identification for effective biometric security”, Publication of Elixir Advanced Engineering Informatics-35(2011)-pp.2830-2832.
29.M.Sundararajan, ”Optical Instrument for correlative analysis of human ECG and Breathing Signal” Publications of International Journal of Biomedical Engineering and Technology- Vol. 6, No.4, pp. 350-362 (2011). ISSN 1752-6418.(Ref. Jor-Anne-I
30..M.Sundararajan, C.Lakshmi & D.Malathi, “Performance Analysis Restoration filter for satellite Images” Publications of Research Journal of Computer Systems and Engineering-Vol.2, Issue-04- July-December-2011-pp 277-28
31. C. Cordeiro, S. Das, and D. Agrawal, “COPAS: Dynamic contention-balancing to enhance the performance of tcp over multi-hop wireless networks,” in Proc. of IC3N, Miami, USA, Oct. 2003, pp. 382–387.