Comparative Performance Evaluation of TPA over TCP Protocol in MANETs

[1]D. Srinivasa Rao, [2]P. Lavanya Kumari
[1,2]Anil Neerukonda Institute of Technology & Sciences, Visakhapatnam, India

Abstract

Providing efficient transport services over multi-hop ad hoc networks is a fundamental building block for this wireless technology. Existing network transport protocols such as TCP and UDP have limitations when fitting into new technologies includes, e.g., increased heterogeneity and mobility. The TCP protocol exhibits poor performance in multi-hop Mobile Ad Hoc Networks (MANETs).The ultimate reason for this is that MANETs behave in a significantly different way from traditional wired networks, like the Internet, for which TCP was originally designed. In particular, route failures and route changes due to node mobility may be frequent events in MANETs. Furthermore, congestion phenomena in MANETs are essentially different from traditional wired networks .In this paper we propose a novel transport protocol – named TPA specifically tailored to the characteristics of the MANET environment.

Unlike other proposals, our protocol is not a modification of the TCP but is specifically tailored to the characteristics of the MANET environment. It is based on a completely new congestion control mechanism, and designed in such a way to minimize the number of useless transmissions and, hence, power consumption. Furthermore, it is able to manage efficiently route changes and route failures. We then compare TCP and TPA through field tests, in terms of throughput and total number of transmitted segments. Our experimental results show that, in the analyzed scenarios, TPA always outperforms TCP in a significant way both in terms of throughput and energy consumption. We show through ns2 based simulations that TPA outperforms both default TCP and TCP-ELFN.

Keywords

Ad Hoc Networks, Mobility, Transport Protocols, TCP, UDP

I. Introduction

The traditional transport protocols such as Transmission Control Protocol (TCP) have served the Internet well for decades. Though it still works fine nowadays and probably will carry on in near future, it appears more strained to fit into the new environment [7]. For example, the congestion control mechanisms in TCP work well in wired networks but often over-react in wireless networks where packets can be lost due to factors other than congestion. Another example is in multimedia applications sharing networks; we need congestion control without ordered reliable delivery, which is not implemented by TCP or User Datagram Protocol (UDP). Such defects appear more and more obviously since new environment coming and being used, includes, e.g., increased heterogeneity and mobility [5, 6]. The lack of appropriate guarantees or specific features has led to the widespread development of specialized protocols used in conjunction with or instead of standard transport protocols. Ad hoc wireless networks pose a big challenge for transport layer protocol and transport layer protocols designed for wired networks like TCP are not suitable for ad hoc wireless networks. There are many issues [1] listed below that a transport layer protocol needs to take into account.

A. Induced Traffic

Ad hoc wireless networks use multi-hop radio relaying, and a link-level transmission affects neighbor nodes of both sender and receiver of the link. This induced traffic affects throughput of the transport layer protocol.

B. Induced Throughput Unfairness

Some MAC protocols, like IEEE 802.11 DCF, may add throughput unfairness to the transport layer. A transport layer protocol needs to take this into account to provide a fair throughput for contesting flows.

C. Separation of Congestion Control, Reliability, and Flow Control

The throughput may be improved if the transports control protocol handles congestion control, reliability and flow control separately. Congestion is usually a local activity that affects only neighboring nodes while reliability and flow control are end-to-end issues. Separation of these should not produce significant control overhead.

D. Misinterpretation of Congestion

Commonly used methods of detecting the congestion by measuring packet loss and retransmission timeout are not suitable for ad hoc wireless networks. Packet loss occurs in wireless networks relatively frequently for several reasons. Bit error rates are much higher than in wired networks and path breaks occur frequently because nodes are constantly moving and they may fail e.g. after draining a battery. Thus, a better method for detecting congestion must be used.

E. Completely Decoupled Transport Layer

In wired networks, transport layer is usually almost completely decoupled from lower network layers. In wireless networks, cross-layer interaction would help transport layer protocol to adapt to the changes in the network.

F. Power and Bandwidth Constraints

Ad hoc wireless networks are constrained by available power and bandwidth. These constraints affect the performance of transport layer protocol.

G. Dynamic Topology

Topology of ad hoc wireless network may change rabidly and this leads to path breaks and partitioning of network. A transport layer protocol should be able to adapt to these changes.

In this paper we propose a novel transport protocol, named TPA (Transport Protocol for Ad hoc networks), and specifically tailored to the characteristics of the MANET environment. It provides a reliable, connection-oriented type of service and includes several innovations with respect to the legacy TCP protocol. In particular, the TPA is able to manage situations that may arise due to nodes' mobility (e.g., route failures and route changes). Furthermore, the congestion control mechanism is completely re-designed with respect to the legacy TCP. Finally, the TPA implements a novel retransmission policy aimed at reducing the number of useless retransmissions and, hence, energy consumption. The rest of the paper is organized as follows in Section 2, we discuss the shortcomings of the TCP design in the context of the ad-hoc network environment. In Section 3, we outline the design goals and key components of TPA. In Section 4, we compare TPA with TCP and TCP-ELFN in the form of performance evaluation under simulated environment and finally in Section 5, we conclude the paper.

II. TCP in MANET

A. Problems of Using TCP over MANETS

The problems that arise with TCP, when used in MANETs. A list of the major factors that particularly affect TCP performance in MANETs is provided. These factors are listed below:

1. Mobility: The mobility of nodes causes routes to change and disconnect frequently which leads to low route stability and availability.

2. High Bit Error Rate (BER): The use of the wireless channel is vulnerable to errors due to weather conditions, obstacles and interference.

3. Unpredictability and Variability: The time-varying nature of wireless channel quality creates uncertainty, which causes substantial difficulty in measuring the RTT and estimating a proper timeout value.

4. Contention: The use of the shared wireless channel limits the ability of a node to send packets. Nodes within a local neighborhood have to compete for wireless channel access. Therefore, the bandwidth obtained by a node depends on the sending need of its neighbors.

Due to the above factors, the performance of TCP is greatly degraded. Mobility of nodes and high BER of the wireless channel cause TCP to misinterpret route failures and wireless errors as network congestion. Moreover, contention will reduce the throughput and fairness of TCP. Sudden delay spikes that are caused by the unpredictability and variability of the wireless channel, also contribute to the poor performance of the TCP.

B. Reliable Transportation in TCP

- Window based transmission
- Slow-Start
- Loss Based Congestion Detection
- Linear Increase Multiplicative Decrease
- Dependence on Acks

1. Window Based Transmissions: Flow control: rwind. How many more packets can the receiver handle?

- Congestion control: cwind
  How many more packets can the network handle?
- Sending rate is determined by min(rwind, cwind)

2. Slow Start

Exponential growth to available capacity. It takes several RTT (Round Trip Time) periods to reach available bandwidth. Slow start is not a serious problem for wire line network. Associations are expected in the congestion avoidance phase for most of the time. Not good for wireless ad hoc network due to the dynamic nature of ad hoc network frequent packet losses, frequent timeouts, more slow start phases. Again, packet loss does not necessarily mean congestion. During the lifetime of an association, considerable amount of time is spent in slow start phase under-utilization of network resources.

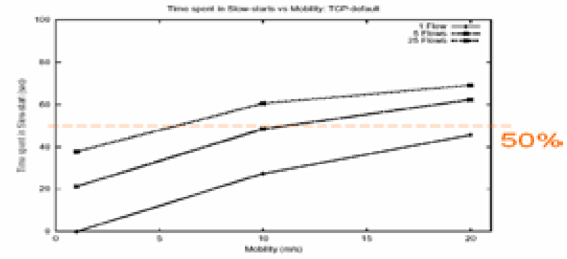


Fig. 1: Time Spent in Slow-Start Phase

In Fig. the time spent in Slow-Start increases with the increasing of the mobility.The proportion of time goes above 50% for the higher load situation. The connections spend a large portion of the lifetime probing for the available bandwidth.

3. Loss Based Congestion Indication

TCP detects congestion through the occurrence of losses either three duplicate ACKs or timeout congestion is by far the main source of packet losses in wireline network. Losses in ad-hoc networks can occur due to either congestion or route failures loss on wireless links means try harder, loss on wired means back off.

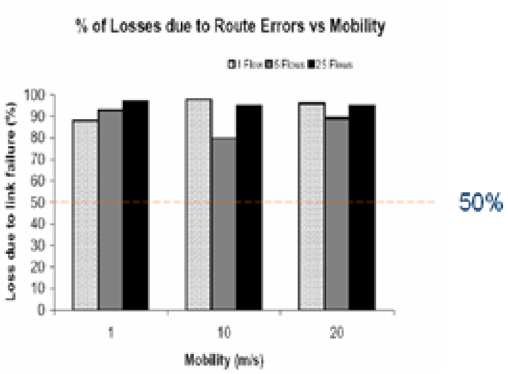


Fig. 2: Losses due to Route Failure

TCP [2] is reliable, end-to-end, connection oriented transport layer protocol. TCP is responsible for congestion control, flow control, in-order delivery of packets, and reliable transportation of packets TCP handles the congestion control in the following way.

TCP regulates the number of packets sent to the network by changing the size of the congestion window. Initially, at the beginning of the TCP session, the size of congestion window is set to one maximum segment size (MSS). If the acknowledgment (ACK) is received during the retransmission timeout period (RTO), size of the congestion window is doubled until the size reaches slow-start threshold. After the slow-start threshold is reached, congestion window increases linearly, by one MSS for every received ACK. If the ACK is not received in time, TCP assumes that the packet is lost and invokes congestion control mechanism: slow-start threshold is halved and the size of congestion window is decreased to one MSS.

- New variable: target window size CongestionThreshold
- Estimate network capacity
- When CongestionWindow reaches CongestionThreshold switch to additive increase
  Initial values
- CongestionThreshold = 8

- CongestionWindow = 1

Loss after transmission 7

- CongestionWindow currently 12
- Set Congestionthreshold = CongestionWindow/2
- Set Congestion Window = 1

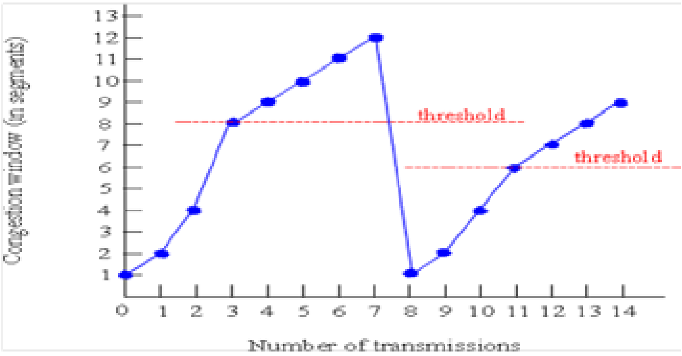


Fig. 3: Loss Based Congestion Indication

The congestion handling is the biggest single issue that makes the traditional TCP a poor choice for ad hoc wireless networks. TCP has been designed for wired networks that have very low bit error rates, thus when the packet loss is encountered, TCP assumes that there is significant problem in the network and initiates aggressive congestion control. In ad hoc wireless networks packet loss can frequently occur for many reasons. First, the bit error rates are generally much higher in wireless network. Second, the nodes are constantly moving and path breaks can occur frequently. If the packet loss occurs frequently, the size of congestion window in TCP will stay at very low level most of the time and this naturally decreases the throughput of the network significantly. In addition, after a route reconfiguration, new route may accept higher throughput. However, TCP does not take this into account. The problem worsens when the path length increases, since the increased path length.

4. Multiplicative Decrease

Multiplicative decrease on congestion window when TCP detects congestion. In MANET, losses can happen by route failure a new route might be used instead slow start to reach available bandwidth TCP-ELFN freezes the TCP sender while a new route is being calculated, but still uses the old congestion windows state after freezing congestion window state for previous route is not appropriate for the new route.

5. Dependence on ACKs

TCP relies on ACK very much the acknowledgement of the correct receiving. The progression of its congestion window Acks can amount to 10-20% of data stream rate large volume of ACKs introduces more contention in MAC layer if the same path is used as reverse path. Large volume of ACKs increases the probability of experiencing route failures (loss of ACKs) if different path is used.

III. TPA in MANET

Ad hoc transport protocol for Ad hoc networks (TPA) is a protocol designed for ad hoc wireless networks; it is not based on TCP. TPA differs from TCP in many ways: TPA uses coordination between different layers, TPA uses rate based transmissions and assisted congestion control and finally, congestion control and reliability are decoupled in TPA. Like many TCP variants, TPA also uses information from lower layers for many purposes like estimating of initial transmission rate, congestion detection, avoidance and control, and detection of path breaks. TPA obtains network congestion information from intermediate nodes, while the flow control and reliability information are obtained from the TPA receiver.

The TPA uses a timer-based transmission where the rate is dependent on the congestion in the network. As packets travel through the network, intermediate nodes attach the congestion information to each TPA packet. This congestion information is expressed in terms of weighted average of queuing delay and contention delay experienced by the packets at intermediate node. The TPA receiver collates this information and sends it back to the sender in the next ACK packets, and the TPA sender can adjust its transmission rate based on this information. During the establishment of the connection, the TPA sender determines the initial transmission rate by sending probe packet to the receiver. Each intermediate node attaches network congestion information to the probe packet and the TPA receiver replies to the sender with an ACK packet containing relevant congestion information. In order to minimize control overhead, TPA uses connection request and ACK packets as probe packets.

TPA increases the transmission rate only if the new transmission rate (R) received from the network is beyond a threshold (x) greater than a current rate (S), e.g. if R>S(1+x) then the rate is increased. The transmission rate is increased only by a fraction (k) of the difference between two rates, i.e.: S=S+(RS)/k, this kind of method avoids rapid fluctuations in the transmission rate.

If the TPA sender has not received ACK packets for two consecutive feedback periods, it significantly decreases the transmission rate. After a third such period, connection is assumed to be lost and the TPA sender moves to the connection initiation phase where it periodically generates probe packets. When a path break occurs, the network layer sends an explicit link failure notification (ELFN) packet to the TPA sender and the sender moves to the connection initiation phase.

A. Key design elements of TPA

- Cross Layer Coordination
- Rate Based Transmissions

-This is the core of TPA

- Decoupled Congestion Control and Reliability

1. Cross Layer Coordination

TPA uses lower layer information and explicit feedback from intermediate nodes to assist the transport layer operations. This information includes (a) an initial rate feedback for a quick-start, (b) a regular rate-based feedback from intermediate nodes to control the sending rate, and (c) a path failure notification for the detection of a route failure.

2. Rate Based Transmissions

TPA uses rate-based transmissions instead of window based transmissions that is used with the conventional TCP. TPA not only uses the rate feedback from intermediate nodes to control the transmission rate but also uses a transmission scheduler to schedule the transmissions evenly over time to reduce the burstiness of the connection.

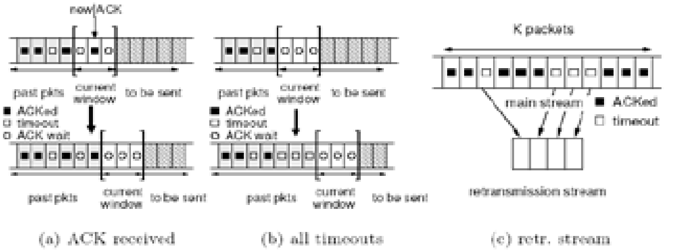
3. Decoupling of Congestion Control and Reliability

In contrast to conventional TCP, TPA decouples congestion control and reliability. TPA does not require the arrival of ACKs to clock out packet transmissions but depends on the regular feedback from the network to perform the congestion control. For reliability, TPA does not employ cumulative ACKs but solely relies on the use of selective acknowledgment (SACK) information that is periodically reported by the receiver to identify packet losses.

The TPA protocol is based on a sliding-window scheme where the window size varies dynamically according to the flow control and the congestion control algorithms (like the TCP protocol [2]). The TPA tries to minimize the number of (re) transmissions in order to save energy. To this end, packets to be transmitted are managed in blocks, with a block consisting of K packets. Specifically, the source TPA grabs a number of bytes corresponding to K TPA packets from the transmit buffer, encapsulates these bytes into TPA packets, and tries to transmit them reliably to the destination. Only when all packets belonging to a block have been acknowledged the TPA takes care to manage the next block. Each packet header includes a sequence number field that identifies the block to which the packet belongs, and a data bitmap field consisting of K bits to identify the position of the packet within the block. The TPA header also includes two fields for piggybacking ACKs into data packets: acknowledgement number and ack bitmap. The acknowledgement number identifies the block containing the packet(s) to be acknowledged, while a bit set in the ack bitmap indicates that the corresponding packet within the block has been received correctly by the destination. Please note that it is possible to acknowledge more than one packet by setting the corresponding bits in the bitmap.

Packet transmissions are handled as follows. Whenever sending a packet, the source TPA sets a timer and waits for the related ACK from the destination. Upon receiving an ACK for an outstanding packet the source TPA performs the following steps: i) evaluates the new window size according to the congestion and flow control algorithms; ii) shifts the window forward, so that it starts with the packet next to the last acknowledged one; and iii) sends packets included in the current window (see Fig. 1-a). On the other hand, if all timeouts related to packets in the current window expire, the source TPA still executes steps i)-iii) above, just as in the case the last outstanding packet has been acknowledged (see Fig. 1-b). In other words, the TPA performs a transmission round during which it tries to send all packets within the block, without retransmitting missed packets.

After the first one, the sender performs a second round for retransmitting packets in the block not yet acknowledged, which are said to form a retransmission stream" (see Fig.1-c). Again, this stream is managed according to steps i)-(iii) above. If a packet within the retransmission stream is acknowledged before being retransmitted, it is dropped from the stream. This procedure is repeated until all packets within the original block have been acknowledged by the destination.



(a) ACK received (b) all timeouts (c) retr. stream

Fig. 4: (a, b) Management of the Sender Sliding Window and (c) Management of the Retransmission Stream

The proposed scheme has several advantages with respect to the retransmission scheme used in the TCP. First, the probability of useless retransmissions is reduced since packets for which the ACK is not received before the timeout expiration are not retransmitted immediately (as in the TCP protocol). This is particularly important in MANETs where nodes are highly mobile and, thus, the timeout value might not reflect the current RTT of the connection. It should also be observed that the longer waiting time in the TPA protocol does not result in a throughput degradation since during this time interval the sender transmits other packets. Second, the TPA is resilient against ACK losses because a single ACK is sufficient to notify the sender about all missed packets in the current block. Third, the sender does not suffer from the out-of-order arrivals of packets. This implies that the TPA can operate efficiently also in MANETs using multi-path forwarding [4], where, on the contrary, the TCP performs very poorly [3].

B. Events of TPA

The TPA protocol also includes some mechanisms to dynamically adapt to the network conditions. Specifically, it is able to detect and manage four kinds of events:

- Route Failure Management
- Route Changes Management
- Congestions Control Mechanism
- ACK Management

1. Route Failure Management

The TPA protocol relies on the network-layer support to detect route failures.

- If ELFN message is enabled then upon receiving ELFN, it goes into freeze state and sets cwnd to 1 segment. It is assumed that no route reestablishment message is supported. So periodically it probes the network for route reestablishment and upon receiving ACK for it, it leaves the freeze state, sets cwnd to cwndmax and starts sending the data.
- If ELFN message is disabled then Sender can still detect route failure by number of consecutive timeouts. If number of timeouts is more than certain number thROUTE then it enters into freeze state.

2. Route Change Management

Similarly to the TCP, the TPA protocol estimates the RTT of the connection and, then uses this estimate to set the retransmission timeout. Both parameters are derived in the same way as in the TCP protocol, ie.,

$\mu RTT\ (n) = g.RTT(n)+(1-g).\ \mu RTT$

$\sigma RTT\ (n) -h.|RTT\ (n) - \mu RTT\ (n)|+(1-h).\ \sigma RTT\ (n-1)$

$Timeout\ (n) = \mu RTT\ (n) + 4.\ \sigma RTT\ (n)$

When μRTT (n) and σRTT (n) are, respectively, the average value and standard deviation of the RTT estimated at the n-th step, RTT(n) is the n-th RTT sample, Timeout (n) is the retransmission timeout computed at the n-th step and, finally, g and h (0<g, h<1) are real parameters [3]. When a route change occurs, packets typically experiences a variation in the RTT and the retransmission timeout might be no longer appropriate for the new path. To avoid possible useless retransmissions the TPA protocol must detect route changes as soon as they occur, and modify the RTT estimation method accordingly. In practice, the TPA detects that a route change has occurred either i) when a new route becomes available after an ELFN; or ii) when th RC consecutive samples of the RTT are found to be external to the interval [μRTT σRTT, μRTT | σRTT]. Upon detecting a route change, the TPA replaces the g and h values in the μRTT and σRTT estimators to greater values g1 and h1 so that the new RTT estimates is heavily influenced by the new RTT immediately after the route change has been detected. Finally, after nRC updates of the estimated RTT, the parameter values are restored to the normal values g and h.

3. Congestion Control Mechanism:

- If ELFN is enabled then Node may fail to send data due to two reasons:

1. Intermediate node may not be able to send data packets to neighbor node called data inhibition.
2. Intermediate node may not be able to relay ACK packets to neighbor node called ACK inhibition that node sends ELFN back to sender or receiver based on above cases whenever sender detects thCONG number of consecutive timeouts, it assumes ACK inhibition and enters congested state and leaves the congested state whenever it receives thCONG number of consecutive ACKs. In this case congestion and route failure can be distinguished by TPA sender.

- Else if ELFN message is not enabled then sender the only way to detect packet loss is consecutive timeout expirations and in that case thCONG = th ROUTE and freeeze = congested state as TPA sender cannot differentiate between route failure and congestion.

Difference between TCP and TPA in terms of congestion window size is that TPA congestion window size is very small (around 2 or 3 segments). In normal conditions, congestion window size is set to maximum congestion window size. In congested state, congestion window size is set to 1 in order to remove congestion.

4. ACK Management

Uses delayed-ACK mechanism. When TPA sender transmits with maximum window size receiver sends an ACK every other segment received else sends an ACK each segment received. Uses txstatus flag to announce window size to receiver. TPA is used with modified version of delayed ACK: in which when sender sends an ACK for each segment if cwnd is 1 otherwise it sends ACK every cwndmax segments.

IV. Performance Evaluation

Simulation Environment

- ns2 simulator
- 1000m*1000m square, 100 nodes
- Random way point mobility with 3 speeds
  1 m/s, 10 m/s, 20 m/s
- DSR routing
- IEEE802.11b MAC
- Network load: 1 flow, 5 flows and 25 flows, resp.
- Packet size: 512 bytes
- Epoch time: 1 sec

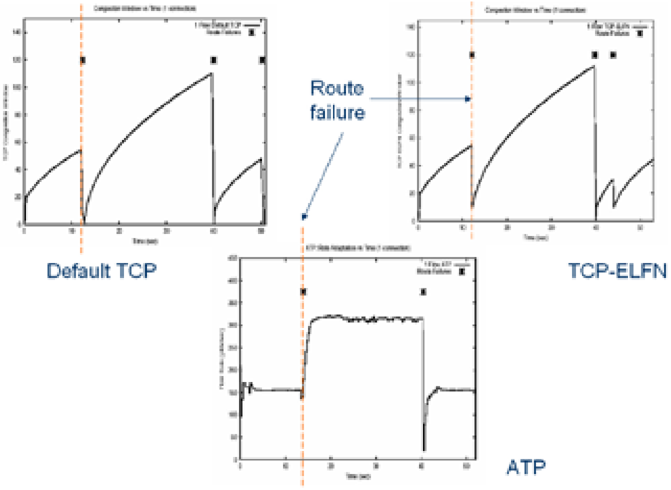


Fig. 5: Congestion window/rate progression vs. time (1 flow)

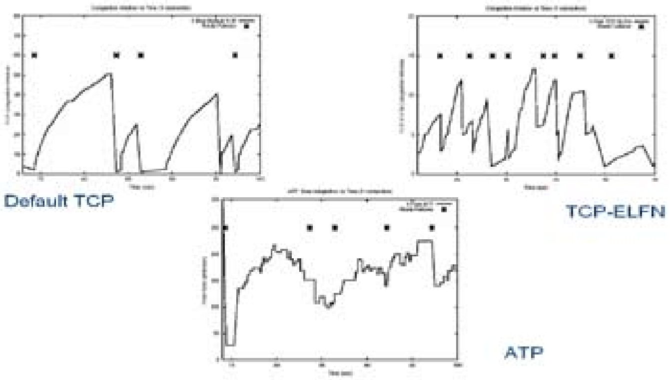


Fig. 6: Congestion window/rate Progression vs. Time (25 flow)

- TPA does not decrease its rate on route failures unless indicated by the rate feedback mechanism
- Owing to quick-start, it quickly catches upto the available bandwidth
- Once it reaches available capacity, it maintains a steady rate

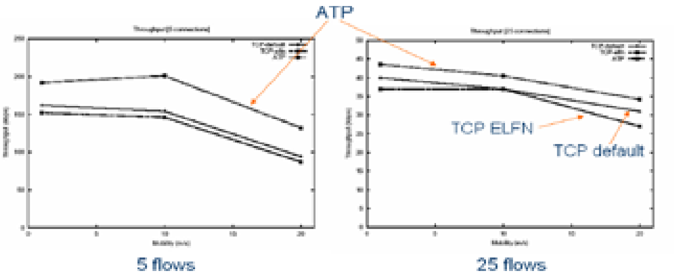
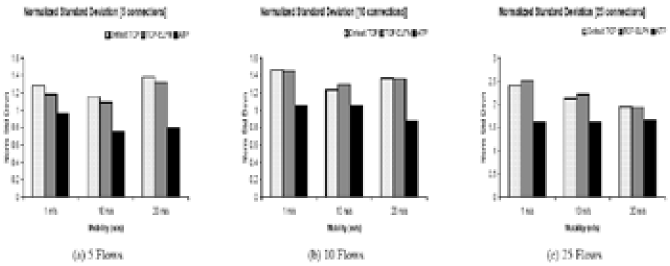


Fig. 7: Throughput vs. Mobility



(a) 5 Flows (b) 10 Flows (c) 25 Flows

Fig. 8: Normalized Standard Deviation

V. Conclusion

The behaviour of TCP over ad-hoc networks is studied extensively in this paper. We infer from the results that a majority of the components of TCP are not suitable for the characteristics of ad-hoc networks. Various reasons are discussed, and the insights gained from the study are used to motivate a new transport protocol called TPA, which is better suited for ad-hoc networks. The protocol addresses all the problems that TCP faces when deployed over ad-hoc networks, and thus shows considerable performance improvement over TCP and TCP-ELFN.

References

[1] C. Murthy, B. S. Manoj. "Ad hoc Wireless Networks: Architectures and Protocols". 2004.

[2] W.R. Stevens, "TCP/IP Illustrated", Vol. 1, Addison Wesley, 1994.

[3] J. Liu and S. Singh, "ATCP: TCP for Mobile Ad Hoc Networks", IEEE J-SAC,Vol. 10, No. 7, July 2001.

[4] V.D. Park, M.S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks", Proceedings of IEEE INFOCOM '97, Kobe, Japan, 1997.

[5] G.Anastasi, E.Ancillotti, M.Conti, A.Passarella."Experimental analysis of a transport protocol for ad hoc networks (tpa)". In PE- WASUN '06: Proceedings of the 3rd ACM international workshop on Performance evaluation of wireless ad hoc, sensor and ubiquitous networks, pages 9–16, New York, NY, USA, 2006. ACM.

[6] H. Elaarag. "Improving TCP performance over mobile networks". ACM Comput. Surv., 34(3):357–374, 2002.

[7] P. G. Bridges, G. T.Wong, M. Hiltunen, R. D Schlichting, M. J. Barrick. "A configurable and extensible transport protocol". IEEE/AC Trans. Netw., 15(6): pp. 1254–1265, 2007.

D. SRINIVASA RAO MCA, M.Phil and M.Tech, is working as an Assistant Professor in the Department of Computer Science Applications at Anil Neerukonda Institute of Technology & Sciences, Visakhapatnam, India. He has 13 years of teaching experience. His area of interest in Computer Networks, Network Security, Image Processing. He has guided more than 50 Post Graduate Students. He has published 2 books, presented 2 papers in National Conferences, 1 paper in international Conference and has attended 20 National Workshops / FDP / Seminars etc., He is a life member of Professional Society like ISTE.

P. LAVANYA KUMARI MCA, is working as an Assistant Professor in the Department of Computer Science Applications at Anil Neerukonda Institute of Technology & Sciences, Visakhapatnam, India. Now she is pursuing M.Tech in Computer Science & Technology with Computer Networks as specialization from Andhra University, India. She has 7 years of teaching experience. Her area of interest in Computer Networks & Network Security. She has attended 4 workshops. She has guided more than 30 Post Graduate Students.