

An Efficient Ad Hoc Transport Protocol (ATP) for MANETs

Kriti Gupta¹, Mansi Gujral² and Nidhi³

¹Department of computer science, Amity University, Noida, UP -201303, India

² Department of computer science, Amity University, Noida, UP -201303, India

³ Department of computer science, Amity University, Noida, UP -201303, India

Abstract

Mobile ad hoc networks (MANETs) are very intricate accumulation of wireless mobile nodes that can be liberatingly self-organized into dynamically ephemeral network topologies. Many researchers spent a considerable duration on amending the performance of reliable transport protocol i.e. TCP for MANETs. TCP surmises that any losses in the network are due to congestion and responds with congestion control mechanism in all situations which results in poor performance. Ad hoc transport protocol- ATP is an incipient protocol that fixates on impotency of TCP for MANETs.

In this paper, we are evaluating the TCP performance under MANET scenario and represents the ineptly ill-fittedness of TCP for MANETs with increasing no. of mobile nodes it responds in decreasing throughput. And discuss the properties and mechanism of ATP. The experiment is done using NS-2 simulation tool. Based on theoretical analysis and experimental result ATP is well suited for MANETs.

Keywords— MANETs, TCP, ATP, ACKs, RTO

1. Introduction

Mobile Ad-hoc networks are benign for a wide number of applications, such as battlefield surveillance, rescue operation and emergency response. Unique characteristics such as lack of infrastructure, shared channel, mobility and limited bandwidth of MANETs differentiate them for traditional wired network. These networks are prone to losses due to link loss, node mobility, and faulty nodes, etc. Traditional transport protocol, TCP [1, 2] was pristinely designed to deal with congestion only. Wireless mobile ad hoc network postulates that all transmission losses are due to congestion and TCP act in response with congestion

control and avoidance algorithms, resulting in degraded performance in wireless systems.

Several works have fixated on ameliorating performance through felicitous modifications to the mechanisms utilized by TCP [3-8]. However it is more fruitful to utilize an incipient transport protocol optimized for MANETs rather than acclimating TCP to the ad hoc environment. The properties and mechanisms of TCP including window predicated transmission, congestion detection, slow-start, multiplicative decrease of congestion window and retransmission, duplicate ACKs, are not fundamentally felicitous for a mobile ad hoc networks [9].

An incipient transport protocol ATP [9] (ad-hoc transport protocol), on the other hand is focused toward the characteristics of ad hoc networks. ATP consists of rate predicated transmissions, route switching and quick-start during connection initiation, no retransmission timeouts, network supported congestion detection and control, decoupled congestion control and reliability.

The rest of the paper is organized as follows: In Section II, we briefly describe the major weak points of TCP in context to MANETs. Section III provides an overview of newly proposed reliable ad hoc transport protocol (ATP). Section IV describes the mechanism of working of ATP protocol. In Section V we show some experiment and result. In the end we conclude with plan for future work.

2. TCP over MANETs

Ad hoc networks pose some tough challenges to TCP due to the fact that it was not designed to operate in such a highly dynamic topology scenario. TCP performance degrades in MANETs because it assumes all the packet losses are due to congestion but most of the link failures are due to mobility. In this section we will discuss the characteristics of TCP over MANETs like (i) window based transmission, (ii) slow-start, (iii) congestion detection, (iv) linear increase and

multiplicative decrease (LIMD), (v) dependence on ACKs.

2.1 Window Based Transmission

TCP is a window based transmission control protocol that uses the sliding-window protocol [10] mechanism. A TCP window is the amount of unacknowledged data a sender can send on a particular connection before it gets an acknowledgment back from the receiver, that it has received some of the data. After receiving the acknowledgement from the receiver, indicating the sequenced number of last well-received packet, the sender slides its window to right side. This mechanism works well for wired scenario but in wireless ad hoc network window based transmission cause burstiness. TCP is self-clocking i.e. reception of ACK triggers next packet transmission. In the absence of timer, several ACKs arrive frequently at sender and even in congestion avoidance phase sender will transmit burst of packets. Bunch of ACKs arrive together is quite common in MANET due to short-term unfairness of the CSMA/CA mechanism [11]. The burstiness results in:

- Varying RTT estimates cause RTO inflation which potentially leads to delayed loss recovery.
- Short-term (induced load) unfairness cause data stream and ACK stream of TCP connection to capture of channel for short period, thus increasing the load on underlying channel higher than average offered load.

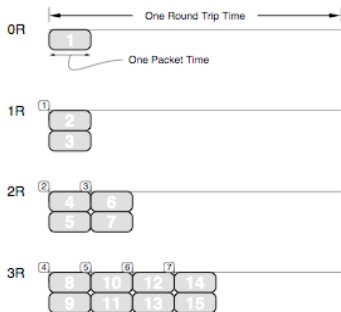


Figure 1: TCP self-clocking example

2.2 Slow-Start

Slow start is a fundamental mechanism of TCP used by sender to control transmission mechanism. It is also known as sender based flow control. It is responsible to work by observing the rate at which sender transmit the new packets into the network is the rate at which the acknowledgments are returned by receiver. During connection establishment phase of TCP, slow start adds a congestion window “cwnd” and initialize is to segment one. When acknowledgments are sent by the receiver, the “cwnd” increases by one segment for each

acknowledgement sent. The sender’s transmission window size is set to minimum of the size of congestion window “cwnd” and the advertised window “awnd”. The sender initiates the transmission by sending one segment and waiting for its ACK. When ACK is returned from receiver side, the “cwnd” is incremented from one to two, and then two to four and so on. We can see from fig. 2 for each of segments that are acknowledged, the “cwnd” grows exponentially.

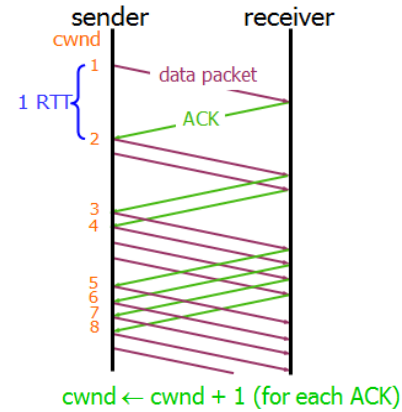


Figure 2: Slow-Start

Thus TCP responds with transmission of two data packet for each ACK this will lead to the problem of burstiness we have discussed. Two other problems associated with the slow-start mechanism in the context of ad-hoc networks are:

- Under-utilization of network resources: Initializing the “cwnd” to segment one yield a very low initial sending rate. In order to quickly grow to true available bandwidth, TCP sender takes several RTT values [12]. Slow start cause a serious problem to ad hoc networks, because of their dynamic nature, connections are suffer from frequent losses which in turn result in frequent timeouts and hence more slow-start phases. Connections spend most of their lifetimes in checking for the available bandwidth instead of operating at the available bandwidth.
- Unfairness: Connections operating in slow start phase violates the TCP’s fairness properties. TCP assumes packet losses are due to congestion, hence it handles the situation with congestion avoidance algorithm which means entering the slow start phase.

2.3 Congestion Indication

TCP uses occurrence of losses as congestion indication. Expiration of retransmission timer and reception of three duplicate ACKs implies the sender that a network

congestion situation is occurring. The sender promptly sets its transmission window to one half of the current window size (i.e. the minimum of the congestion window and the receiver's advertised window size). If congestion is indicated by a timeout, sender enters the Slow Start phase. If congestion is indicated by three duplicate ACKs, sender invokes Fast Retransmit and Fast Recovery algorithms. Congestion is mainly occur due to packet losses in wired networks, but same is not the case with ad hoc wireless networks. In traditional wireless networks, random wireless channel error rates also contribute to losses, and considerable amount of research has been done toward either hiding such losses from TCP through link-layer reliability, or enhancing TCP with mechanisms that can distinguish congestion losses from random errors [13, 14]. In ad hoc networks, congestion and random wireless errors therewith mobility plays primary role in losses cognize by connections, hence treating losses as congestion indication turns out to be inappropriate.

2.4 Linear Increase and Multiplicative Decrease

Linear increase and multiplicative decrease (LIMD) algorithm of TCP control the congestion by making changes in the congestion window size. It linearly grows congestion window size "cwnd" when congestion goes down and exponentially decreases the "cwnd" when congestion is goes up. The approach taken is to increase the transmission rate (window size, "cwnd"), checking for available bandwidth, until loss occurs. The notion of linear increase is to increase the "cwnd" by a fixed amount every round trip time (RTT), this the slow start phase. When congestion is detected, the sender decreases the transmission rate by a multiplicative factor; e.g. divide the cwnd by two whenever timeout occurs. If timeout happens continuously, the "cwnd" is decremented to 1. This is because the minimum initialized value for "cwnd" is 1. When there is no congestion is detected, the sender increases the "cwnd" by one after every successful ACK received. LIMD is not appropriate for ad hoc wireless network. It performs slow start to reach available bandwidth and hence more vulnerable to route failure before true bandwidth is achieved. TCP freezes the current state as soon as route fails and resume as soon as new route found [3, 4, 5] but uses old congestion window state of previous route for new route which is inappropriate.

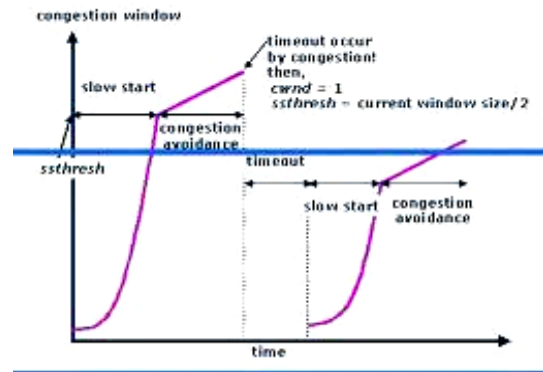


Figure 3: Linear Increase and Multiplicative Decrease of Congestion Window

2.5 Dependence on ACKs

Reducing the dependence on ACKs in TCP is very difficult to achieve because TCP uses acknowledgement for reliable delivery, packet loss and to perform effective congestion control. Dependence on ACKs results in following two problems:

- If forward path and reverse path are same, ACKs can amount to 10-20% of data stream rate and large volume of ACKs introduces more contention in MAC layer.
- If forward and reverse path are different then large volume of ACKs increases the probability of experiencing route failures (loss of ACKs).

3. Overview of ATP

In this section we will discuss the key design elements of ATP.

3.1 Cross Layer Coordination

Cross layer design is widely used for wireless networks. Cross layer protocol interaction, when used appropriately, can lead to increased network efficiency and better QoS support in MNAETs. Cross layer mechanism is necessary for wireless networks because state of physical medium changes over time. ATP uses cross layer coordination between different layers of the protocol stack. ATP uses layer coordination for path failure notification and initiating sending-rate estimation for the new route.

3.2 Rate Based transmission

In rate based transmission a maximum transmission rate and an average transmission rate is assigned to each source. The network assigns rates such that probability of occurrence of congestion is less, by knowing the information of burstiness and desired rates

of all nodes [15]. This method basically provides a promised rate without dependence on feedback from the intermediate nodes, and thus is more appropriate than windows for use on high-speed networks. ATP using rate based transmission transmits fixed sized packets in each interval of time. Thus, the need for self-clocking by the arrival of ACKs is eliminated and this allows decoupling of congestion control mechanism from the reliability mechanism. ATP avoids drawbacks due to burstiness.

3.3 Decoupling of Congestion Control and Reliability

Ad hoc transport protocol (ATP) decouples the congestion control from reliability control. Whereas in TCP dependence on ACKs makes congestion control and reliability control tightly coupled. ATP uses feedback from intermediate nodes to provide congestion control and using coarse grained receiver feedback that eliminates dependence on ACKs to ensure reliability. For congestion control intermediate nodes provides feedback of available rate, this feedback is piggybacked in the forward path and receiver is responsible for collating feedback information collected from intermediate node to sender to adjust the sending rate accordingly. For reliability receiver uses selective acknowledgement (SACK) to notify the sender explicitly the list of packets or segments that have been acknowledged so far. In TCP selective acknowledgement (SACK) and cumulative ACK are complementary policies while ATP is based purely on SACK policy.

3.4 Quick-Start

During connection initiation and when timeout occurs each time TCP enters in to slow start phase, thus spending most its lifetime in doing slow start and this will degrade the network performance. Unlike TCP, ATP uses quick start mechanism in lieu of slow start and takes single round trip time to reach the available bandwidth, whereas TCP uses several RTT values to reach true available bandwidth. ATP performs quick start during connection initiation and when network undergoes changes or new route is selected. Estimating the true available bandwidth once again when new path is used prevent over-utilization and under-utilization of network resources.

4. ATP Mechanism

In this section we will describe the mechanism used by ATP protocol. It primarily consists of mechanism at intermediate node, at sender and at receiver. ATP sender is responsible for connection management, initial rate estimation, congestion control and reliability

whereas ATP receiver is responsible for reliability, flow control, collated congestion messages. Intermediate nodes play an important role in providing congestion feedback to sender. In rest of the section we will discuss in details all the three mechanism.

4.1 Role of Intermediate Node

Intermediate nodes measures the sum of two parameters i.e. average queuing delay (Q_i) per packet at node itself and the average transmission delay (T_i) at node's transmitter. These two parameters are maintained on per node basis. For each transmitted packet intermediate nodes update and record the values of Q_i and T_i .

$$Q_i = \alpha * Q_i + (1 - \alpha) * Q_s$$

$$T_i = \alpha * T_i + (1 - \alpha) * T_s$$

Here Q_s and T_s are the queuing delay and transmission delay experienced by the transmitted packet. We use a value of 0.75 for α . ATP has another rate feedback field D at each packet (note $1/D$ is the actual rate). For every transmitted packet intermediate nodes checks the value of D , if it is smaller than $Q_i + T_i$ then update the value of D to $Q_i + T_i$. This D field indicates the receiver maximum average delay packet experienced at each intermediate node.

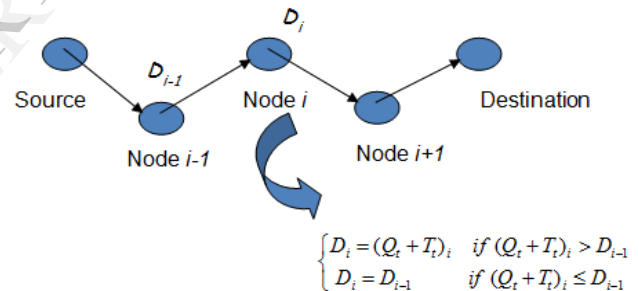


Figure 4: Role of Intermediate Node

4.2 ATP Receiver

ATP receiver runs an epoch timer of E period in order to send rate feedback, reliability feedback and flow control feedback periodically to sender. ATP receiver receives feedback from intermediate nodes through information given in the D field on the packets and sends it to sender.

Rate Feedback

For every incoming packet in a flow, the receiver calculates average delay of the D field specified in the packet and sends it to the sender [9].

$$D_{avg} = \beta * D_{avg} + (1 - \beta) * D$$

4.2.1 Reliability Feedback

The ATP receiver uses selective ACKs (SACKs) to sender for providing feedback containing information

about any loss recognized in the data stream. Unlike TCP-SACK that uses only 3 blocks per ACK, ATP uses large no. SACK, at most 20 SACK blocks in its reliability feedback.

4.2.2 Flow Control Feedback

Unlike in TCP, where the flow control is achieved through appropriate window advertisements, ATP performs flow control by observing the application rate R_{app} . R_{app} is the rate at which the application is processing in sequence data from the receive buffer. Flow control feedback is provided to sender when application rate (R_{app}) is smaller than the rate feedback and the rate feedback information is replaced with the R_{app} [9].

$$\begin{cases} D_{avg} & \text{if } D_{avg} \geq 1/R_{app} \\ 1/R_{app} & \text{otherwise} \end{cases}$$

4.3 ATP Sender

The ATP sender consists of most the transport layer protocol mechanisms. ATP sender has following functionality: (i) quick-start, (ii) congestion control, and (iii) reliability.

4.3.1 Expeditious-Start

To find the initial sending rate ATP sender sends a probe packet to the receiver, after 1RTT ATP sender receives rate feedback from receiver and adjusts the available sending rate accordingly. Expeditious start is performed during connection establishment and when new path is setup due to link failure.

4.3.2 Congestion Control

Unlike TCP, which uses two-phase congestion control mechanism with an increase-phase and a decrease-phase, ATP uses a three phase congestion control mechanism with increase, decrease, and maintains phases. One of the key differences between TCP's congestion control mechanisms and that of ATP's is the network feedback that ATP's mechanisms use [9]. ATP sender updates the sending rate S during Increase, Decrease and Maintain Phase

$$\begin{cases} S = S + \frac{R-S}{k} & \text{if } S < R - \phi \cdot S & \text{Increase phase} \\ S = R & \text{else if } S > R & \text{Decrease phase} \\ \text{unchange} & \text{otherwise} & \text{Maintain phase} \end{cases}$$

In case of the loss of feedback packets ATP addresses:

- Multiplicative decrease of sending rate if feedback does not appear in every epoch time.
- If no feedback till the end of the third epoch, send again a probe packet to the receiver.

4.3.3 Reliability

ATP sender ensures reliability by processing the SACK information that identifies the packet to be retransmitted. It gives preference to data for retransmission rather than new data.

5. Experiment and Results

For simulation experiment, a wireless scenario is created by writing tcl script in ns2 for different no. of mobile nodes and throughput of received packets for each node is calculated. The calculated average throughput is for TCP connection between the mobile nodes in ad hoc wireless network which results in poor network performance, as with increasing number of mobile nodes links are break and packets are dropped drastically due to their random mobility.

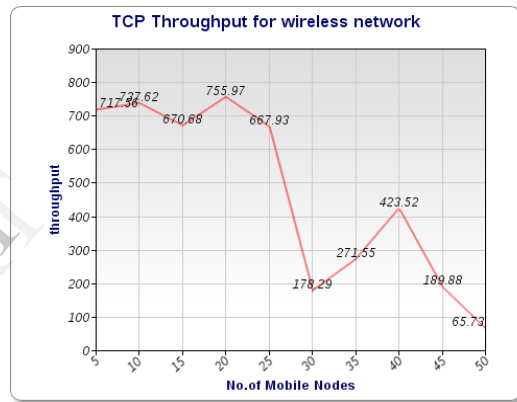


Figure 5: TCP average throughput of received packets under wireless scenario.

In fig. 4, using TCP connection for ad hoc network average throughput of received packets is consistently decreasing and at point it dropped drastically. TCP is failed to distinguish between congestion losses and mobility losses. It has been observed that based on theoretical analysis if ATP will be used instead of TCP, ATP will overcome all the limitations of TCP faced in ad hoc network. Because ATP uses rate based transmission instead of the window based transmission performed by TCP and it decouples the congestion control mechanism.

Table 1: Comparison between TCP and ATP

Transmission Protocol	Ad Hoc Transport Protocol
Transmission control protocol is a window based protocol.	Ad hoc transport protocol is rate based protocol.

TCP relies on feedback from receiver only. ATP relies on feedback from receiver as well as from intermediate nodes.

In TCP SACK information is complementary to cumulative ACK. ATP relies totally on SACK information.

In TCP congestion control and reliability mechanism are tightly coupled due to dependence on ACKs. ATP congestion control and reliability mechanism are decoupled this eliminates the dependence on ACKs.

TCP uses layered architecture. ATP uses cross layered architecture.

When path failure occur TCP freezes the sender state but uses the old congestion window state of pervious route for new route. ATP freezes the sender state and enters in to the route establishment phase.

Slow start of TCP results in under-utilization of network resources. Quick start of ATP prevents under-utilization of network resources.

6. Conclusion and Future Prospects

The demeanor of TCP for mobile ad hoc network is studied in detail in this project. It is been concluded that TCP is infelicitous for mobile ad hoc networks. The result shows that TCP poorly perform in mobile ad hoc network and many reasons have been discussed that point out major limitation of TCP over mobile ad hoc network. This motivates to the utilization of incipient protocol specially designed for mobile ad hoc network i.e. ATP. Theoretical analysis prognosticates that ATP is felicitous for wireless mobile ad hoc network as it surmounts the limitations of TCP.

In future performance of ATP will be evaluated under different network scenario and instead of TCP, ATP will be utilized. Throughput and delay for different network scenario will be computed. More research will be done to show the efficacy of ATP for Mobile ad hoc networks.

7. References

- [1] J. B. Postel. Transmission Control Protocol. RFC, Information Sciences Institute, Marina del Rey, CA, September 1981. RFC 793.
- [2] W. R. Stevens. TCP/IP Illustrated, Volume 1. Addison-Wesley, Reading, MA, Nov 1994.

- [3] Chandran, K., Raghunathan, S., Venkatesan, S., and Prakash, R. (2001) A Feedback Based Scheme for Improving TCP Performance in Ad Hoc Wireless Networks. IEEE Personal Communication Magazine, 8, 34-39. Special Issue on Ad Hoc Networks.
- [4] Holland, G. and Vaidya, N. (2002) Analysis of TCP performance over mobile ad hoc networks. Wireless Networks, 8, 275-288.
- [5] Liu, J. and Singh, S. (2001) ATP: TCP for Mobile Ad Hoc Networks. IEEE Journal on Selected Areas in Communications, 19, 1300-1315.
- [6] Sun, D. and Man, H. (2001) ENIC - An Improved Reliable Transport Scheme for Mobile Ad Hoc Networks. IEEE Globecom Conference, San Antonio, TX, November, pp. 2852-2856. IEEE Computer Society Press.
- [7] Wang, F. and Zhang, Y. (2002) Improving TCP Performance Over Mobile Ad-hoc Networks With Out-of-Order Detection and Response. MobiHoc '02: Proceedings of the 3rd ACM International Symposium on Mobile Ad hoc Networking & Computing, New York, NY, USA, June, pp. 217-225. ACM Press.
- [8] Kim, D., Toh, C., and Choi, Y. (June 2001) TCP-BuS: Improving TCP Performance in Wireless Ad Hoc Networks. J. Commun. and Net., 3, 175-186.
- [9] Sundaresan, K., Hsieh, V. A. H., and Sivakumar, R. (2005) ATP: A Reliable Transport Protocol for Ad Hoc Networks. IEEE transactions on mobile computing, 4,588-603.
- [10] A. S. Tanenbaum, Computer Networks, Pentice Hall, Second Edition,1988.
- [11] C. E. Koksal and H. Balakrishnan, "An Analysis of Short-term Fairness in Wireless Media Access Protocols (poster)," in Proceedings of ACM SIGMETRICS, Measurement and Modeling of Computer Systems, Santa Clara, CA, 2000, pp. 118-119.
- [12] W. Stevens. TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms, January 1997, RFC 2001.
- [13] H. Balakrishnan, S. Seshan, E. Amir, and R. Katz, "Improving TCP/IP performance over wireless networks," in Proceedings of ACM MOBICOM, Berkeley, CA, Nov. 1995.
- [14] P. Sinha, N. Venkitaraman, R. Sivakumar, and V. Bharghavan, "WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks," in Proceedings of ACM MOBICOM, Seattle, WA, Aug. 1999.
- [15] Jane M. Simmons "Analysis of Rate Assignments in a Rate-Based Flow Control Scheme", M.S.thesis, Massachusetts Institute of Technology, February 1990.