An Enhanced Packet Retransmission Method for Improving TCP-Aware Source Routing in Mobile Ad-Hoc Network

[1]S. Kannan, [1]S. Karthik and [2]V.P. Arunachalam
[1]Department of Computer Science and Engineering,
SNS College of Technology, Sathy Main Road, Coimbatore, 641035 Tamil Nadu, India
[2]SNS College of Technology, Coimbatore, Tamil Nadu, India

Abstract: Mobile Ad-hoc Networks (MANET) are self-organizing and self-configuring multihop wireless networks where the structure of the network changes dynamically. This is mainly due to the mobility of nodes. The nodes in the network not only acts as hosts but also as routers that route data to or from other nodes in network. In mobile ad-hoc networks, a routing procedure is always needed to find a path so as to forward the packets appropriately between the source and the destination. In a MANET, temporary link failures and route changes occur frequently. With the assumption that all packet losses are due to congestion, TCP performs poorly in such an environment. This study proposes a new mechanism called TASR, TCP-aware source routing which can improve TCP performance in wireless Ad-hoc networks. TASR adds a hold state to an existing routing protocol to reduce consecutive timeouts, retransmissions and out-of-ordered packets in TCP. In the simulation study, TASR achieves up to a 60% improvement in performance without requiring any TCP stacks in end systems to be modified.

Key words: Ad-hoc network, performance evaluation, routing protocol, DSR, TCP, India

INTRODUCTION

The Internet Engineering Task Force (IETF) created a Mobile Ad-hoc Network (MANET) working group to standardize IP routing protocol functionality suitable for wireless routing application within both static and dynamic topologies with increased dynamics due to node motion and other factors. The vision of Ad-hoc networks is wireless internet where users can moveanywhere anytime and still remaining connected with the rest of the world.

A mobile Ad-hoc network is a network in which a group of mobile computing devices communicate among themselves using wireless radios without the aid of a fixed networking infrastructure. Due to its dynamic property, mobile Ad-hoc networks have gained a lot of attention lately as a way of providing continuous network connectivity to mobile computing devices in various areas.

However because the topology of networks changes dynamically, route changes frequently and failure to find a valid route promptly would result in a significant drop in performance. When TCP is used as

transport protocol such a drop in performance is rather expected because packets sent to an invalid route are all lost. Specifically, TCP performance can suffer due to the following reasons:

- Packet loss due to broken routes can result in the counterproductive invocation of TCP's congestion control mechanisms
- Selecting an invalid alternative path while reestablishing a broken one, this would result in consecutive timeout
- Longer RTO (Retransmission Time Out) value which results from consecutive timeout
- Out of ordered TCP segments

So, there have been a lot of research to address the routing problem in mobile Ad-hoc networks (Holland and Vaidya, 2001; Chandran *et al.*, 1998; Ahuja *et al.*, 2000; Dyer and Boppana, 2001; Zhang and Wang, 2002).

Routes are broken frequently as the movement of mobile terminals gets faster. Frequent route change makes TCP to have multiple packets losses. TCP has only one way to recover from multiple packets losses and

Corresponding Author: S. Kannan, Department of Computer Science and Engineering, SNS College of Technology,
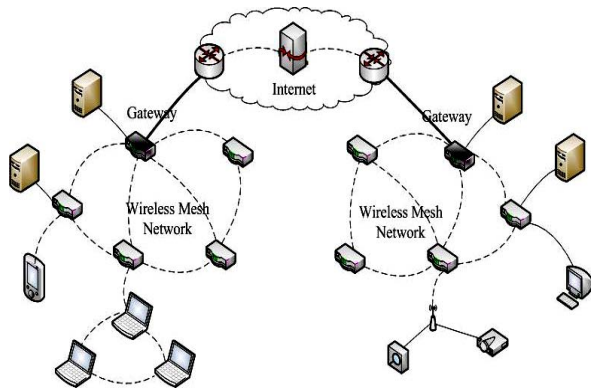Sathy Main Road, Coimbatore, 641035 Tamil Nadu, India

Fig. 1: MANET approach

that is to expire the retransmission timer. But a consecutive timeout makes the RTO (Retransmission Timeout) value exponentially back off (Stevens, 1994) and longer RTO reduces link utilization and decreases throughput dramatically (Vaidya, 2001). So, we think that consecutive timeout is a key factor that affects the TCP performance (Fig. 1).

MANET characteristics: The fundamental difference between fixed networks and MANET is that the computers in a MANET are mobile. Due to the mobility of these nodes, there are some characteristics that are only applicable to MANET. Some of the key characteristics are described below (Perkins and Royer, 1999):

Dynamic network topologies: Nodes are free to move arbitrarily, meaning that the network topology which is typically multi-hop may change randomly and rapidly at unpredictable times.

Bandwidth constrained links: Wireless links have significantly lower capacity than their hardwired counterparts. They are also less reliable due to the nature of signal propagation.

Energy constrained operation: Devices in a mobile network may rely on batteries or other exhaustible means as their power source. For these nodes, the conservation and efficient use of energy may be the most important system design criteria.

The MANET characteristics described above imply different assumptions for routing algorithms as the routing protocol must be able to adapt to rapid changes in the network topology.

RELATED WORK

Recent studies have addressed the TCP performance problems caused by route failures in a mobile Ad-hoc network. Since, TCP assumes that all packet losses are due to network congestion, although the cause of packet losses is route failure, TCP performs congestion control. Because this behavior is the major reason by which it shows dramatic drop in TCP performance, many studies try to distinguish between route failure and network congestion and thereby improve the performance of the routing protocols (Karthik *et al.*, 2008a, b).

Chandran *et al.* (1998) proposed a feedback-based scheme called TCP-Feedback or TCP-F. In this scheme, when an intermediate node detects route failure, it explicitly sends a RFN (Route Failure Notification) message to the TCP sender. On receiving the RFN, the TCP sender suspends packet transmissions and freezes all states in cluding the RTO value and CWND (the size of Congestion Window).

When an intermediate node learns of a new route to the destination, it sends a RRN (Route Reestablishment Notification) message to the TCP sender which then restores its previous state and resumes transmission. The conclusion of the study was that the average route repair time has a major impact on TCP performance.

ROUTING PROTOCOLS IN MANET

There are different criteria for designing and classifying routing protocols for wireless Ad-hoc networks. For example, what routing information is exchanged when and how the routing information is exchanged when and how routes are computed etc.

Proactive vs. reactive routing: Proactive schemes determine the routes to various nodes in the network in advance, so that the route is already present whenever needed. Route discovery overheads are large in such schemes as one has to discover all the routes. Examples of such schemes are the conventional routing schemes, Destination sequenced Distance Vector (DSDV) (Johnson and Maltz, 1996). Reactive schemes determine the route when needed. Therefore, they have smaller Route discovery overheads.

Single path vs. multi Path: There are several criteria for comparing single-path routing and multi-path routing in Ad-hoc networks. First, the overhead of route discovery in multi-path routing is much more than that of single-path routing (Haas and Pearlmane, 2001). On the other hand, the frequency of route discovery is much less in a

network which uses multi-path routing, since the system can still operate even if one or a few of the multiple paths between a source and a destination fail. Second, it is commonly believed that using multi-path routing results in a higher throughput.

Table driven vs. source initiated: In table driven routing protocols, up-to-date routing information from each node to every other node in the network is maintained on each node of the network. The changes in network topology are then propagated in the entire network by means of updates. Destination Sequenced Distance Vector Routing (DSDV) and Wireless Routing Protocol (WRP) are 2 schemes classified under the table driven routing protocols head. The routing protocols classified under Source Initiated On-Demand Routing, create routes only when desired by the source node (Karthik *et al.*, 2009a, b). When a node requires a route to a certain destination, it initiates what is called as the route discovery process. Examples include DSR and AODV.

Dynamic Source Routing (DSR) protocol: Dynamic Source Routing (DSR) is a routing protocol for wireless mesh networks. It is similar to AODV in that it forms a route on-demand when a transmitting computer requests one. However, it uses source routing instead of relying on the routing table at each intermediate device. Many successive refinements have been made to DSR including DSRFLOW.

Determining source routes requires accumulating the address of each device between the source and destination during route discovery. The accumulated path information is cached by nodes processing the route discovery packets. The learned paths are used to route packets. To accomplish source routing, the routed packets contain the address of each device the packet will traverse. This may result in high overhead for long paths or large addresses, like IPv6. To avoid using source routing, DSR optionally defines a flow id option that allows packets to be forwarded on a hop by hop basis.

This protocol is truly based on source routing whereby all the routing information is maintained (continually updated) at mobile nodes. It has only two major phases which are Route discovery and route Maintenance. Route reply would only be generated if the message has reached the intended destination node (route record which is initially contained in route request would be inserted into the route reply).

To return the route reply, the destination node must have a route to the source node. If the route is in the destination node's route cache, the route would be used. Otherwise, the node will reverse the route based on the route record in the route reply message header (this requires that all links are symmetric). In the event of fatal transmission, the route maintenance phase is initiated whereby the route error packets are generated at a node. The erroneous hop will be removed from the node's route cache; all routes containing the hop are truncated at that point. Again, the route discovery phase is initiated to determine the most viable route (Ahuja *et al.*, 2000).

TCP-AWARE SOURCE ROUTING

Existing routing protocols (Johnson and Maltz, 1996; Perkins and Royer, 1999; Haas and Pearlman, 2001; Corson and Park, 2001) in Ad-hoc networks are designed to reestablish broken routes as soon as possible. However, since there is a delay between when a route is broken and when the source node is informed of the route failure, the packets sent through the broken route will be lost. If UDP is the transport protocol such burst packet losses would not cause a serious problem in performance. When the source node has alternative routes, the source node keeps sending the packets out through one of them. If the chosen alternative route is again invalid, TCP will suffer a series of consecutive timeouts.

If timeouts occur contiguously, the RTO value of TCP's retransmission timer is exponentially backed off. Since, all packet transmissions are suspended until the timer is expired, link utilization is significantly reduced.

A key observation is that invalid routes cause consecutive timeout. The chance of invalid routes is quite high because the source node depends on the routing information from its neighbors that are not updated with the route failure. The chance will be higher when mobile terminals move faster in mobile Ad-hoc networks (Karthik *et al.*, 2008c).

The congestion control of TCP does not help improve the performance because too many packets are lost and the timer expiration is the only way to start retransmission.

Therefore, we need a mechanism that prevents the alternative routes from being invalid when the source node is informed of a route failure. So, we propose the mechanism called TASR that consists of adding a hold state and refreshing alternative routes in its routing table. Here is how TASR works.

When the source node is informed of a route failure, TASR checks whether the transport protocol is TCP or not. If it is, TASR makes the routing protocol transits to the hold state. Hold state is the state in which the routing protocol does not forward any data packets. In the hold state, TASR starts probing alternative route and such probing is performed in parallel. In order to get the fastest

path, the destination node should reply to such probing packets as soon as possible. Finally, when TASR gets the fastest route by n-parallel probing, it escapes from its hold state and starts packet transmission by using that path (Karthik *et al.*, 2009a, b).

An advantage of TASR is that it does not require the modification of TCP stack. In mobile Ad-hoc network, it is accepted that mobile terminals need a new routing protocol because of the unique nature of Ad-hoc networks. But it is desirable that the transport layer stays independent of the underlying networks. To the best of the knowledge all the previous routing protocols in Ad-hoc network required TCP stack modification and TASR is the first attempt that achieves no modification.

DESIGN AND IMPLEMENTATION OF TASR

Add a hold state to DSR (Johnson and Maltz, 1996), one of the existing routing protocols in order to implement TASR. The reason we choose DSR is that 1) it is widely accepted in Ad-hoc networks, 2) its architecture fits the framework of TASR. Although, we implement TASR by adding a hold state on DSR, we believe that TASR can be applied to any other routing protocol (Karthik *et al.*, 2008b, c).

The DSR protocol is an on-demand routing protocol that is based on the concept of source routing. The protocol consists of 2 phases: route discovery and route maintenance. When a mobile node wishes to send a packet to a destination node, it first checks its route cache to determine whether it already has a route to the destination node. If the mobile node does not have such a route, it initiates route discovery by broadcasting a ROUTE REQUEST packet. When either the destination node or an intermediate node that contains a valid path to the destination in its route cache, receives the route request Packet, a ROUTE REPLY packet is generated. As the source node receives ROUTE REPLY, route discovery is done and DSR initiates packet transmission. Route maintenance is accomplished through the use of ROUTE ERROR packets. ROUTE ERROR packets are generated at a node when the data link layer encounters a fatal transmission problem. Nodes that receive ROUTE ERROR packets remove these invalid paths from their route cache.

In order to implement TASR, we define 2 additional control packets: ROUTE PROBE and REPLY PROBE. The ROUTE PROBE packet is used to probe fresh paths when the source node receives a ROUTE ERROR packet. And REPLY PROBE packet is used to reply to the ROUTE PROBE packet.

TASR consists of probing routes in n-parallel and adding a hold state. When DSR reports ROUTE ERROR, TASR does not immediately select an alternative path in the route cache but transits to a hold state and start parallel probing. The probing process is as follows:

- TASR selects n paths from the route cache (select 3, heuristic value)
- TASR sends a ROUTE PROBE message to each path
- Destination nodes which have received a ROUTE PROBE, send REPLY PROBE messages using reverse paths
- When the first REPLY PROBE message is received, TASR releases its hold state and resumes packet transmission using that path. It is because the path that has the shortest round trip time is the best path at that given moment

TASR has two ways to escape from the hold state and return to the DSR state. Normally, it transit to hold state→select state→DSR state. However, if all ROUTE PROBE or REPLY PROBE messages are lost in immediate nodes, TASR may wait in the hold state infinitely. So, introduce the TASR timer that is used to back to the DSR state. This TASR timer uses timeout value (TTO) be set to:

$$TTO = TCP's\ RTO + \alpha \qquad (1)$$

where, α is a delay variance factor with a recommended value of 2. Since, TCP's SRTT adapt itself to mobile Ad-hoc networks slowly in a way (Kim and Noble, 2001), we add α, a delay variance factor, to RTO. The transition diagram of TASR is shown in Fig. 2.
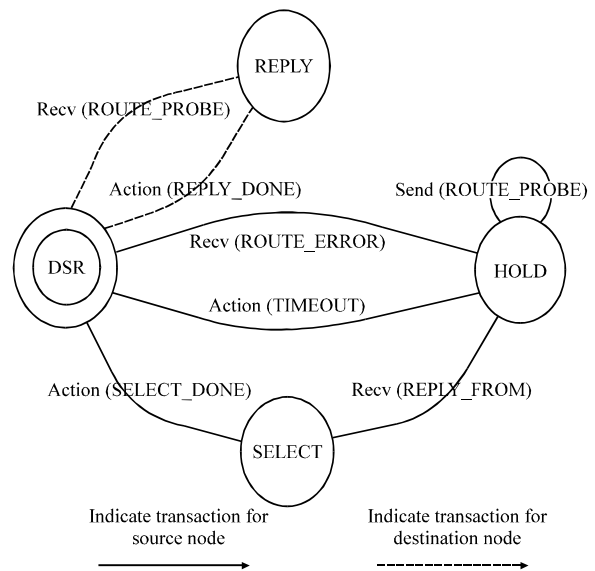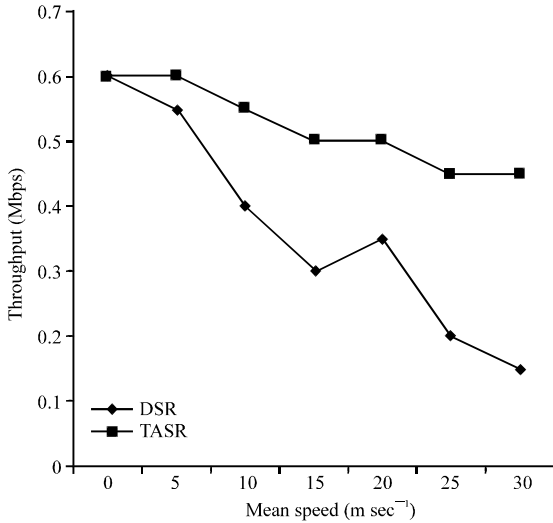

Fig. 2: The transition diagram of TASR

Fig. 3: Comparison of TCP performance based on routing protocol (DSR,TASR)

The simulation study is done in the NS2 network simulator (Fall and Varadhan, 1997). NS-2 is a discrete event simulator that was developed as part of the VINT project at the Lawrence Berkeley National laboratory.

The extensions implemented by the CMU Monarch project (CMU Monarch Group, 1998) enable it to simulate mobile nodes connected by wireless network interfaces. We extended the NS-2 DSR protocol implementation to include TASR. All results are based on a network configuration consisting of TCP-Reno over IP on an IEEE 802.11MAC layer. The network model consists of 30 nodes in a 1500×300 m flat, rectangular area. Each node uses a wireless channel model with a transmission range of 250 m. The nodes move according to the random waypoint mobility model.

We measured, the throughput of TCP with and without TASR, varying the mean speed from 2-30 m sec^{-1} and got the count of the retransmission timer's backoff in each case. Also, we measured the sum of each mobile node's throughput in case the network has 5 and 10 TCP sessions. The results are shown in Fig. 3.

Figure 3 shows the throughput of TASR compared to DSR. Significant improvements in throughput can be observed in the best case (e.g., when the mean speed is 30 m sec^{-1}). The improvement in throughput is largely due to avoiding consecutive timeouts. The throughput gain increases as the mean speed increases. It is because as the mobility of nodes in mobile Ad-hoc network increases, the probability of route failures gets higher. In the current design, TASR initiates only when the protocol of the transport layer is TCP and the routing protocol of the source node received a ROUTE ERROR message.

CONCLUSION

Normal TCP performs poorly in mobile Ad-hoc networks because of frequent route changes. In the scheme, TASR does not send out packets until it discovers a reliable route. By holding the state of routing protocol, TASR reduces consecutive timeouts, retransmissions and out-of-ordered packets in TCP.

This protocol achieves up to a 60% improvement in performance compared with DSR. Also, it shows more outstanding improvements in performance as the mobility of mobile terminal increases. We also experimented with UDP and got a similar result as TCP. In addition, TASR enhances TCP performance just by modifying the routing protocol without any modification of TCP.

REFERENCES

Ahuja, S., J.P. Singh and R. Shorey, 2000. Performance of TCP over different routing protocols in mobile ad-hoc networks. Proceedings of the Spring Tokyo IEEE 51st Vehicular Technology Conference, May 15-18, Tokyo, Japan, pp: 2315-2319.

CMU Monarch Group, 1998. CMU Monarch extensions to the NS-2 simulator. http://monarch.cs.cmu.edu/cmu-ns.html.

Chandran, K., S. Raghunathan, S. Venkatesan and R. Prakash, 1998. A feedback based scheme for improving TCP performance in ad-hoc wireless networks. Proceedings of the 18th International Conference on Distributed Computing Systems, May 26-29, Amsterdam, Netherlands, pp: 472-479.

Corson, M.S. and V. Park, 2001. Link Reversal Routing. Addison-Wesley, USA.

Dyer, T. and R. Boppana, 2001. A comparison of TCP performance over three routing protocols for mobile Ad-hoc networks. Proceedings of the 2nd ACM International Symposium on Mobile Ad-hoc Networking and Computing, Long Beach, CA, USA., Oct. 4-5, ACM, New York, USA., pp: 56-66.

Fall, K. and K. Varadhan, 1997. NS notes and documentation. The VINT Project, UC Berkeley, LBL USC/ISI and Xerox PARC. http://www-mash.cs.berkeley.edu/ns.

Haas, Z.J. and M.R. Pearlmane, 2001. ZRP: A Hybrid Framework for Routing in Ad-hoc Networks. Addison-Wesley Longman Publishing Co. Inc., Boston, MA, USA., pp: 221-253.

Holland, G. and N. Vaidya, 2001. Analysis of TCP performance over mobile Ad-hoc networks. Proceedings of the ACM International Symposium on Mobile Ad-hoc Networking and Computing, Oct. 4-5, ACM Press, New York, pp: 259-262.

Johnson, D.B. and D.A. Maltz, 1996. Dynamic source routing in Ad-hoc wireless networks. Mobile Comput., 353: 153-181.

Karthik, S., R.M. Bhavatharini and V.P. Arunachalam, 2008a. Analyzing interaction between denial of service (DOS) attacks and threats. Proceedings of the International Conference on Computing, Communication and Networking, Dec. 18-20, IEEE Computer Society, pp: 1-9.

Karthik, S., T. Ravichandran and V.P. Arunachalam, 2009a. Analyzing interaction between denial of service (DoS) attacks and threats. Int. J. Soft Comput., 4: 68-75.

Karthik, S., V.P. Arunachalam and T. Ravicahndran, 2008b. A comparative study of various IP trace back Strategies and simulation of IP trace back. Asian J. Inform. Technol., 7: 454-458.

Karthik, S., V.P. Arunachalam and T. Ravicahndran, 2009b. A novel Direction Ratio Sampling Algorithm (DRSA) approach for multi directional geographical traceback. Int. J. Comp. Sci. Sec., 3: 272-279.

Karthik, S., V.P. Arunachalam and T. Ravichandran, 2008c. Multi directional geographical traceback with in directions generalization. J. Comput. Sci., 4: 646-651.

Kim, M.K. and B. Noble, 2001. Mobile network estimation. Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, July 16-21, ACM, Rome, Italy, pp: 298-309.

Perkins, C.E. and E.M. Royer, 1999. Ad-hoc on-demand distance vector routing. Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, Feb. 25-26, New Orleans, LA., pp: 90-100.

Stevens, W.R., 1994. TCP/IP Illustrated: The Protocol. Vol. 1, Addison Wesley, Reading, MA., pp: 304-306.

Vaidya, N., 2001. Mobile Ad-hoc networks: Routing, MAC and transport issues. ACM MOBICOM 2001, Tutorial.

Zhang, Y. and F. Wang, 2002. Improving TCP performance over mobile adhoc networks with out-of-order detection and response. Proceedings of the 3rd ACM International Symposium on Mobile Ad-hoc Networking and Computing, June 9-11, Lausanne, Switzerland, pp: 217-225.