An Enhanced Scheme for Improving Packet Retransmission over Mobile Ad-Hoc Networks Using TCP

[1]S. Karthik, [1]S. Kannan, [2]V.P. Arunachalam, [3]T. Ravichandran and [1]M.L.Valarmathi
[1]Department of Computer Science and Engineering, SNS College of Technology,
Sathy Main Road, Coimbatore-641035, Tamil Nadu, India
[2]SNS College of Technology, Sathy Main Road, Coimbatore-641035, India
[3]Hindustan Institute of Technology, Pollachi Main Road, Coimbatore-641032, India

Abstract: Denial of Service (DoS) attacks constitutes one of the major threats and among the hardest security problems in today's Internet. Of particular concern are Distributed Denial of Service (DDoS) attacks whose impact can be proportionally severe. With little or no advance warning, a DDoS attack can easily exhaust the computing and communication resources of its victim within a short period of time. Because of the seriousness of the problem many defense mechanisms have been proposed to combat these attacks. This study aims to provide an understanding of the existing attack methods, tools and defense mechanisms, so that a better understanding of DDoS attacks can be achieved. The goal of the study is to simulate an environment by extending NS2, setting attacking topology and traffic which can be used to evaluate and compare the methods of DDoS attacks and tools. Based on the simulation and evaluation results, more efficient and effective algorithms, techniques and procedures to combat these attacks may be developed.

Key words: DDoS, attack methods, tools, defenses and simulation, mobile, India

INTRODUCTION

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 re-establishing a broken one, this would result in consecutive timeout

- Longer RTO (Retransmission Timeout) 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, 1999; 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 that is to expire the retransmission timer. But a consecutive timeout makes the RTO (Re-transmission Timeout) value exponentially back off (Stevens, 1994) and longer RTO reduces link utilization and decreases throughput dramatically (Holland and Vaidya, 1999). So we think that consecutive timeout is a key factor that affects the TCP performance.

MANET characteristics: The fundamental difference between fixed networks and MANET is that the computers in a MANET are mobile.

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

Due to the mobility of these nodes, there are some characteristics that are only applicable to MANET. Some of the key characteristics are described (Zhang and Wang, 2002).

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 (Karthik *et al.*, 2009a).

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. In this study, the researchers explore a new way to enhance TCP performance by minimizing the consecutive timeout in the case of frequent route changes, without modifying TCP stack.

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.*, 2008c). Chandran 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, including 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 Re-establishment 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.

TCP-aware source routing: Existing routing protocols (Johnson and Maltz, 1996; Perkins and Royer, 1999; Haas and Pearlmane, 2001; Corson and Park, 2001) in ad hoc networks are designed to re-establish 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. Also, note that 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.*, 2008a, b). Also note that 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. And 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.*, 2009b). 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 network. 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.

MATERIALS AND METHODS

Design and implementation of TASR: The researchers 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 (it is widely accepted in ad hoc networks) its architecture fits the framework of TASR. Although, we implement TASR by adding a hold state on DSR, TASR can be applied to any other routing protocol (Karthik *et al.*, 2008b). The DSR protocol is an on-demand routing protocol that is based on the concept of source routing. The protocol consists of two 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 two 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 (researchers 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 transist to hold state→select
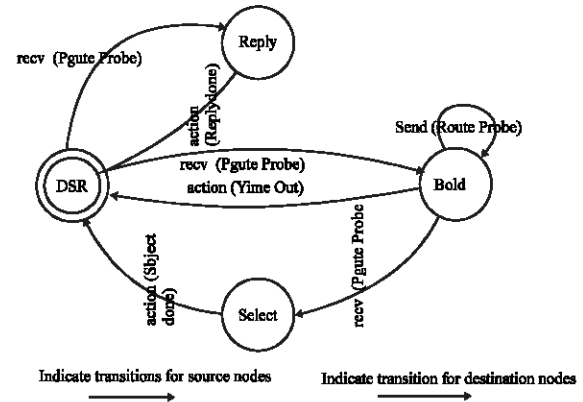

Fig. 1: The transition diagram of TASR

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, we 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. 1.

RESULTS AND DISCUSSION

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 (1998) enable it to simulate mobile nodes connected by wireless network interfaces. We extended the NS-2 DSR protocol implementation to include TASR as described earlier. All results are based on a network configuration consisting of TCP-Reno over IP on an 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 we 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
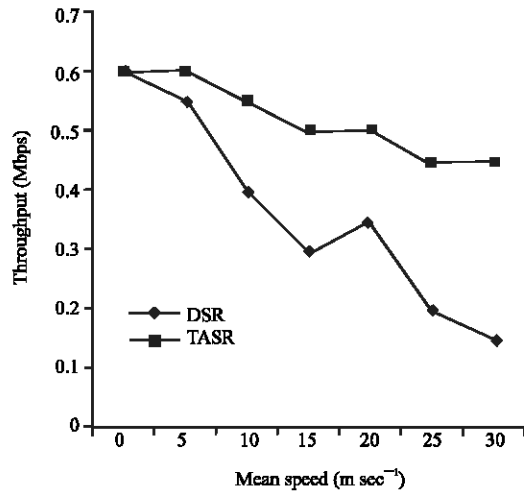
Fig. 2: Comparison of TCP performance based on routing protocol (DSR, TASR)

10 TCP sessions. The results shown in Fig. 2. Figure 2 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. We can see that 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, re-transmissions 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.

ACKNOWLEDGEMENTS

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.

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, 1999. Analysis of TCP performance over mobile ad hoc networks. Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, Aug. 15-19, ACM, New York, USA., pp: 219-230.

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, 2008b. Multi directional geographical traceback with n directions generalization. J. Comput. Sci., 4: 646-651.

Karthik, S., V.P. Arunachalam and T. Ravicahndran, 2008c. 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, 2009a. A novel Direction ratio sampling algorithm (DRSA) approach for multi directional geographical traceback. Int. J. Comp. Sci. and Security, 3: 272-279.

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

Kim, M.K. and B. Noble, 2001. Mobile network estimation. Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, (AICMCN'01), 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 Volume I: The Protocols. Addison-Wesley Professional, USA., ISBN-13: 978-0201633467.

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.