A Survey of TCP over Routing Protocols

Shailesh Kumar Patel[1], R.K.Gupta[2], Madhulika Sharma[3]

Department of Computer Science and Engineering

Azad Institute of Engineering and Technology, Lucknow

shaileshkumarpatel2008@gmail.com[1], kailashcs09@gmail.com[2], madhulikasharma4@gmail.com[3]

ABSTRACT

A Mobile Ad-hoc Network (MANET) is a collection of mobile devices dynamically forming a communication network without any centralized control and pre-existing network infrastructure. Due to the presence of mobility in the MANET, the interconnections between stations are likely to change on a continual basis, resulting in frequent changes of network topology. Consequently, routing becomes a vital factor and a major challenge in such a network. In this paper we study the four IETF (Internet Engineering Task Force) standardized routing protocols on MANETs. The four routing protocols that are considered in the analysis are Ad-hoc On-demand Distance Vector (AODV), Dynamic Source Routing (DSR), Location Aided Routing (LAR1) and Wireless Routing Protocol (WRP). In addition, from a transport layer's perspective, it is necessary to consider Transmission Control Protocol (TCP) as well for MANETs because of its wide application, which enjoys the advantage of reliable data transmission in the Internet. However, the factors such as scalability and mobility cause TCP to suffer from a number of severel performance problems in an ad-hoc environment. Hence, it is of utmost importance to identify the most suitable and efficient TCP variants that can robustly perform under these specific conditions. Therefore, this study also makes an attempt to evaluate the performance of the three TCP variants (Reno, New Reno and SACK) under a variety of network conditions.

Keywords- *MANET, AODV, DSR, LAR1, WRP, TCP.*

1. INTRODUCTION

Mobile network can be classified into infrastructure networks and Mobile Ad-hoc NETworks (MANET). A MANET [1, 3] is a group of wireless mobile computers or nodes, where the nodes are created in the network can change their location from time to time. A mobile ad-hoc network is sometimes also known as infrastructure-less network because mobile ad-hoc networks are automatically self-organized networks without support of infrastructure. A routing protocol is mainly used to discover the shortest, most efficient and correct paths while providing the data transmissions between different wireless devices in ad-hoc network. A MANET is an evolving technology, which offers a cost-effective and scalable method to connect wireless devices. Lately, this technology has become increasingly popular due to its potential application in many domains. For instance, such a network can be helpful in rescue operations where there is not sufficient time or resource to configure a wired network. MANETs are also very useful in military operations.

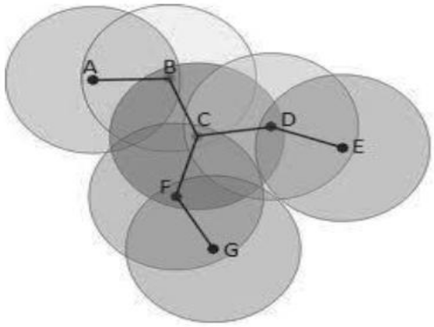

Figure 1.1: Mobile Ad-hoc NETwork

2. TCP PERFORMANCES

2.1 Network function

TCP is a transport layer protocol which hides the rigid IP layer restrictions of maximum packet length and potential packet delivery problems, and delivers a byte stream service where the application knows that all bytes sent to TCP will be delivered at the destination application in the correct order without packet loss. TCP handles retransmission of lost data, rearranges of out-of-order data, and helps minimizes network congestion. TCP [3] focuses on reliable delivery, and this may increase the delivery delay, since it must wait for retransmission of lost message or reorder out-of-order messages. Thus, it is less suitable for traffic that requires low delay, e.g. interactive streaming video conferencing and Voice over IP (VoIP).

The reliability of TCP depends on acknowledgment packets sent from the destination to the source, to confirm to the source that the destination has received the data. The source keeps track of each sent packet, and maintains a window for packets for which it awaits Acknowledgments (ACKs). A new packet is not sent until a slot in this window is available. In addition, a timer is kept from the time the packet was sent, in case a packet disappears or is corrupted. The packet is retransmitted if the timer expires.

2.2 TCP segment structure

TCP receives data from a data stream (from the application). The data are segmented into chunks and with an added TCP header, these accounts for a TCP segment. This TCP segment is transmitted over the network wrapped in an Internet Protocol (IP) datagram. The TCP header can be seen in Figure 2.1 and consists of the following fields:

Source port (16 bits) is the sender's port.
Destination port (16 bits) is the receiver's port.
Sequence number (32 bits) represent either (if the SYN bit is set) the initial sequence number, or (if the SYN bit is not set) the sequence number of the current packet.
Acknowledgment number (32 bits) represent either (if the ACK bit is set) the sequence number of the next expected byte to be received from the sender, or (if the ACK bit is not set) the acknowledgment of the other end's initial sequence number itself.

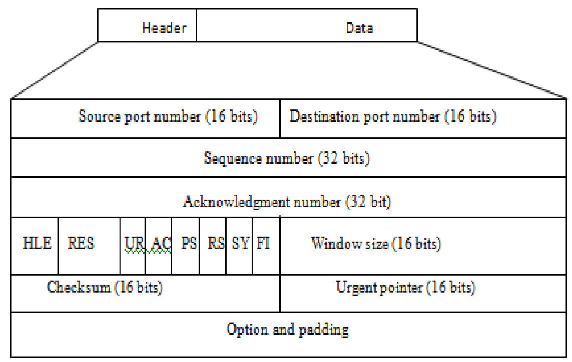

Figure 2.1: TCP header

Data offset (4 bits) specifies the size of the TCP header in the number of 32 bits words.
Reserved (6 bits) is for future use (set to zero).
URG (1 bit) indicates that the Urgent pointer field is valid.
ACK (1 bit) indicates that the Acknowledgment field is valid.
PSH (1 bit) If set, requests the receiver to push the buffered data to the receiving application.
RST (1 bit) Connection reset.
SYN (1 bit) If set means that the sequence numbers should be synchronized between the sender and receiver. It is only set in the first packet sent from each end.
FIN (1 bit) If set means that the sender has finished sending data, and there will be no more data from the sender.
Window size (16 bits) is set by the receiver. Announces the maximum number of bytes that the receiver is currently willing to receive.

Checksum (16 bits) is used for controlling if the header and data contains error.

Urgent pointer (16 bits) represents an offset from the sequence number indicating the last urgent data byte, if the URG bit is set.

Options (0-32 bits) can contain various options, and a padding to ensure the field's length is divisible on 32 bits.

2.3 TCP variants

In this dissertation we analyze the performance of three TCP variants (Reno, New Reno and SACK). These three mechanisms, TCP Reno, TCP New Reno and TCP Selective Acknowledgement (SACK) support fast recovery algorithm.

2.3.1 TCP Reno

Along with the implementation of the basic principles of Tahoe, the TCP Reno version adds more mechanisms so as to detect the lost packets in shorter time and also prevent the pipeline from being empty every time a packet is lost. The packet segment is assumed to be lost as soon as the duplicate acknowledgements are reached to its threshold level. Then the TCP enters the Fast Re-transmit phase through which the lost segment is retransmitted. When the Fast Retransmit phase is completed, TCP Reno employs the Fast Recovery algorithm which does not let the pipeline to be empty and also provides extra incoming duplicate ACKs to clock subsequent outgoing packets. Moreover, Fast Recovery assumes whenever a duplicate ACK is attained, each time there is a single packet left in the pipe. As a result, the TCP Reno sender is capable of making sharp estimation over the amount of outstanding data in the network. Meanwhile, after entering the Fast Recovery phase, the TCP sender waits until half a window of dup ACKs are achieved, and then transmits a new data packet for each additional dup ACK [5]. Finally, the sender leaves the Fast Recovery phase when it receives a new ACK for the new data.

The variant TCP Reno can smoothly detect the single packet drop; however this version experiences difficulty in case of multiple packets dropped from the window and the performance becomes almost as like as Tahoe version. When multiple packets are dropped, the loss information of the initial packet is arrived after the reception of the duplicate ACK. On the other hand, the information about the second packet is obtained after the acknowledgement of the retransmitted initial packet is reached to the sender. Furthermore, this ACK of the retransmitted initial packet is arrived after one RTT and hence it takes longer time to process the second packet loss.

2.3.2 TCP New Reno

In case of multiple packet loss, the TCP New-Reno does not wait for the retransmission timer to be expired and hence this variant provides a dominating performance over the Reno version. In New Reno, the performance concerns about the behavior of the partial ACKs, which do not take TCP out of Fast Recovery phase while it takes TCP out from the Fast Recovery phase in Reno version [5]. Moreover, in New-Reno, receiving partial ACKs often indicates the loss of the packets which instantly follows the acknowledged packet in the sequence space. Thus for the multiple packet losses, the New-Reno becomes able to retransmit all the packets lost from a particular window and therefore the New-Reno does not leave the Fast Recovery phase unless the acknowledgement for all outstanding data in the network is completed.

However, New-Reno may experience poor performance as it takes one RTT for identifying the packet loss and therefore it is possible to infer about the information of other lost packet only when the ACK for the first retransmitted segment is received [4].

2.3.3 TCP SACK

TCP uses a cumulative acknowledgment scheme through which only a single lost segment can be detected per round trip time. Moreover, this scheme does not allow the received packets that are not at the left edge of the receiver window to be acknowledged. Hence in order to discover the lost packet, the sender has to either wait for a roundtrip time or retransmit the received packet unnecessarily. Consequently, TCP loses its ACK-based clock and thus decreases the overall throughput.

In order to overcome these limitations, a SACK mechanism, combined with a selective repeat retransmission policy is developed. TCP SACK is basically an upgraded version of TCP New Reno which takes steps to solve the major problems experienced by the New Reno version. Such problems include the detection of multiple lost packets and re-transmission of more than one lost packet per RTT [2]. With selective acknowledgments, the information about the arrived data segments can be reached successfully to the sender. As a result the sender only needs to retransmit the actual lost packet. The TCP SACK offers a significant feature so that the segments are acknowledged selectively instead of being acknowledged cumulatively. In addition, there is a block present in each ACK which monitors the acknowledgments and reports the sender of which segments have been acknowledged. For increasing and decreasing the congestion window size, the congestion control algorithms of SACK version are found almost same as Reno. The TCP SACK retains the basic properties and services of Tahoe and Reno, for instance, ensures high robustness even in the existence of the out-of-order packets. However, when multiple packets are lost from the data window, the properties between SACK and other variants can be differentiated.

In the Fast Recovery stage of SACK version, a variable is maintained by the sender in order to measure the number of outstanding data in the network. This variable is called a pipe and it is not maintained in any of the earlier TCP versions. As long as the estimated number of outstanding packets is found below than the congestion window value, a data is transmitted or retransmitted by the sender [1]. Moreover, when the sender sends a new data or retransmits an old packet, the variable pipe is incremented by one while it is decremented by the same value upon receiving a duplicate ACK with a selective acknowledgment option.

Though TCP SACK provides many advantages, it is not an easy task to implement selective acknowledgment options in TCP SACK version. Hence, currently the TCP receivers are found to be reluctant for providing the selective acknowledgment option.

3. ROUTING PROTOCOLS

In this study we use four routing algorithms (AODV, DSR, LAR1 and WRP) in a MANET [7] environment.

3.1 Ad-hoc On Demand Distance Vector (AODV)

The Ad-hoc On Demand Distance Vector (AODV) is considered an efficient MANET routing protocol and supports both unicast and multicast routing mechanisms. The AODV routing protocol utilizes an on-demand technique in order to discover the routes. This means that the route between two endpoints (nodes) is formed as per requirement for the source node and maintained as long as the routes are needed. Moreover, the protocol uses a destination sequence number to recognize the most recent path and to guarantee the freshness of the routes. Reactive protocols like AODV shrinks the control traffic overhead at the cost of higher latency in discovering new routes. AODV does not have any function until there is a valid route between the source and destination in MANET. Upon requiring the formation of a new route, the source node transmits a Route Request (RREQ) packet.

After flooding the RREQ packet, the source node waits until a Route Reply (RREP) packet is received as an acknowledgement. However, within a specific time, a RREP may not be received and in that case a new RREQ is to be sent again by the source node. And for this additional transmission of RREQ, the predefined waiting interval needs to provide a binary exponential back-off and therefore it is multiplied by two (2) each time. The binary exponential back-off must be utilized in order to reduce the network congestion. After receiving a RREQ, the neighbor node either generates a RREP message to the sender or rebroadcasts the RREQ depending on the availability of a valid route to the destination. The validity of the route is confirmed after making a comparison between the sequence number of the intermediate node and the destination sequence number of the Route Request packet. Once the RREP is received by the source node, it stores the information of this particular route and starts transmitting data toward that destination. However, in case of the reception of the multiple RREPs, the route with the shortest hop count will be selected.

In case a link failure is occurred, a Route Error (RERR) message is created and returned to the originator of the data in a hop-by-hop manner and the process replicates. The purpose of generating the RERR message is to inform other nodes about the current broken link. The source node disables the route as soon as it receives the Route Error message and invokes the route discovery mechanism again if it is necessary.

3.2 Dynamic Source Routing Protocol (DSR)

The Dynamic Source Routing Protocol (DSR) is a simple and efficient routing protocol designed specifically for use in multi-hop wireless ad-hoc networks of mobile nodes. DSR allows the network to be completely self organizing and self configuring, without the need for any existing network infrastructure. DSR has been implemented by numerous groups, and deployed on several test beds. Network using the DSR protocol have been connected to the Internet. DSR can interoperate with Mobile IP, and nodes using Mobile IP and DSR have seamlessly migrated between WLANs, cellular data services, and DSR[8] mobile ad-hoc networks.

The protocol is composed of the two main mechanisms of "Route Discovery" and " Route Maintenance", which work together to allow to nodes to discover and maintain routes to arbitrary destination in the ad-hoc network.

3.3 Location-Aided Routing Protocol (LAR1)

The Location-Aided routing protocol (LAR) is a reactive (on-demand) routing protocol that uses the location information of the mobile nodes. Location information about nodes is obtained using Global Positioning System (GPS). LAR is advancement over Dynamic Source Routing (DSR) in context of route request packet flooding. In LAR, location information of the mobile nodes are used to flood a route request packet in a forwarding zone only called as request zone instead of the entire ad-hoc network. This request zone is determined by location information of the destination. Routing overhead in an ad-hoc network is reduced by the use of location information; this is one of the advantages of LAR. Complexity of protocol is nullified assuming accurately. A limitation of this protocol is every host requires a GPS device.

LAR defines two different types of request zones: LAR Scheme 1 (LAR1) and LAR Scheme 2 (LAR2).

LAR1 schemes use two zones: Expected zone and Request zone.

Location-Aided Routing (LAR1) routing protocol is an on-demand routing protocol which exploits location information. It is similar to Dynamic Source Routing (DSR) Routing protocol, but with the additional requirement of GPS information. In scheme 1 (implemented), the source defines a circular area in which the destination may be located, determined by the following information:

- The destination location known to the source
- The time instant when the destination was located at that position
- The average moving speed of the destination.

The smallest rectangular area that includes this circle and the source is the request zone. This information is attached to a ROUTE REQUEST by the source and only nodes inside the request zone propagate the packet. If no ROUTE REPLY is received within the timeout period, the source retransmits a ROUTE REQUEST via pure flooding.

3.4 Wireless Routing Protocol (WRP)

The Wireless Routing Protocol (WRP) is proactive unicast routing protocol for mobile ad-hoc networks (MANETs). It is a table-based protocol similar to DSDV that inherits the properties of Bellman Ford Algorithm. The main goal is maintaining among all nodes in the network regarding the shortest distance to every destination. WRP is another loop-free proactive protocol. WRP is path-finding algorithm with the exception of avoiding the count-to-infinity problem by forcing each node to perform consistency checks of predecessor information reported by all its neighbors. Each node in the network uses a set of four tables to maintain more accurate information: Distance table, Routing table, Link-cost table, Message retransmission list table. In case of link failure between two nodes, the nodes send update messages to their neighbors. This eliminates looping situations and enables faster route convergence when a link failure occurs. Loop avoidance is based on providing for the shortest path to each destination both the distance and the second-to-last hop (predecessor) information. Despite the variance in the number of routing tables used, and the difference in routing information maintained in these tables, proactive routing protocols like WRP are distance vector shortest-path based, and have the same degree of complexity during link failures and additions.

4. TCP PERFORMANCES IN MANETS

Even though TCP ensures reliable end-to-end message transmission over wired networks, a number of existing researches have showed that TCP performance can be substantially degraded in mobile ad-hoc network [3]. Along with the traditional difficulties of wireless environment, the mobile ad-hoc network includes further challenges to TCP. The different types of constraints influencing the TCP performance in MANET environment are:

- High BER
- Route Failures
- Path Asymmetry Impact
- Network Partitioning
- Power Scarcity
- Multipath Routing
- Interaction between MAC Protocol & TCP

4.1 High BER

High Bit Error Rate (BER) is caused due to multipath fading, Doppler shift and signal attenuation. This cause TCP data segments to be lost and thereby the congestion control mechanisms are triggered unnecessarily by the TCP sender.

4.2 Route Failures

In MANET, the mobility of the node is considered as the major reason for the route failure and the route reestablishment is needed instantly in case of route failure. However, it is likely that a new route establishment may experience longer duration than the RTO of the sender. In consequence of that, the TCP sender will unnecessary deploy congestion control mechanism.

4.3 Path Asymmetry Impact

The network topology is changed very frequently and arbitrarily within MANETs, which leads to the creation of an asymmetric path. This path formation negatively influences the TCP performance since TCP is highly dependent on time responsive feedback information. The sender starts transmitting data in a burst when a number of ACKs are received together, which causes the packet to be lost. In MANETs, path asymmetry can be grouped into different forms such as loss rate asymmetry, bandwidth asymmetry and route asymmetry.

4.4 Network Partitioning

A network partition takes place when a node departs from the network, resulting in an isolation of some parts of a mobile ad-hoc network. These fragmented portions are defined as partitions. In a MANET environment, TCP considers network partitioning as one of the most imperative challenges which is mainly caused due to mobility or energy-constrained (limited battery power) operation of nodes. When the source and the destination of a TCP connection lie in different parts of the network, all transmitting packets are found to be dropped by the network. As a result, the congestion control algorithm will be invoked instantly by the TCP sender.

Again, the serial timeouts at the TCP sender can be generated in case of frequent disconnections in the network. This may trigger a longer idle period for the network through which the connection can be re-established. However, the TCP does not found to move from the back off state.

4.5 Power Scarcity

Each mobile node carries batteries which have limited power supply; hence the network suffers the node lifetime problem. Each node in

MANET works as a router and an end system, therefore needless retransmissions of the packet cause the consumption of this limited energy resource. As a result, an inefficiency of the available power is utilized.

4.6 Multipath Routing

In order to reduce the frequency of route re-computation, some routing protocols preserve multiple routes between the sender and the receiver. However, this may result in the arrival of a huge number of out-of-sequence packets to the receiver. Consequently, it causes the receiver to generate duplicate ACKs and the sender to employ the congestion control mechanisms [26].

4.7 Interaction between MAC Protocol and TCP

In a MANET environment, the intercommunication between the TCP mechanisms and 802.11 MAC protocol may lead to unexpected severe challenges such as link capture effect, instability, and hop unfairness. The causes of these problems include the hidden station and exposed station problems of the 802.11 MAC protocols.

5. CONCLUSION

In this survey we have studied the various Routing Protocols and TCP variants. Empirical results illustrate that the performance of a routing protocol varies widely across different network sizes and different mobility models and hence the study results from one model cannot be applied to other model. Hence we have to consider the mobility of an application while selecting a routing protocol. Every routing protocol has their own significance. Some routing protocols are better in some condition while they perform badly in other conditions. It is depending upon on which situation we have to communicate each other.

For future work, the implementation of these routing protocols with different parameters can be done in different scenario and conclude which routing protocol is better in which condition.

6. REFERENCES

[1] K.Natrajan, Dr. G. Mahadevan ,” A Comparative Analysis and performance evaluation of TCP over MANET Routing Protococls” in Int Jr of Advanced Computer Engineering and Architecture, “Vol 1,No.1, June 2011.

[2] K. Salah, P. Calyam, and M. I. Buhari, "Accessing readiness of IP networks to support desktop video conferencing using GLOMOSIM," *Journal of Network and Computer Applications*, vol. 10, no. 2007, pp. 1-23, 2007.

[3] Marga Nácher, Carlos T. Calafate, Juan-Carlos Cano, and Pietro Manzoni, "Comparing TCP and UDP performance in manets using multipath enhanced versions of DSR and DYMO," in *Proceedings of the 4th ACM workshop on Performance evaluation of wireless ad hoc, sensor,and ubiquitous networks*, pp. 39 45,October 2007.

[4] I. Chlamtac, M. Conti and J. Liu, “Mobile adhoc networking: imperatives and challenges *,”Ad Hoc Networks Journal*, vol.1, no. 1, pp. 13-64, Jul. 2003.

[5] O. Bazan, U. Qureshi, M. Jaseemuddin and H.M. El-Sayed, "Performance Evaluation of TCP in mobile ad-hoc networks," in *The Second International Conference on Innovations in Information Technology, IIT’05*, Toronto, Canada, 2005, pp. 175-185.

[6] R. Ramanathan and J. Redi, “Brief overview of ad hoc networks: challenges and directions,” *IEEE Communications Magazine*, vol. 40, no. 5, pp. 20-22, May 2002.

[7] D. Johnson, Y. Hu and D. Maltz,"Dynamic Source Routing Protocol (DSR) for mobile ad hoc networks," *RFC 4728 (Experimental)*, IETF, Feb. 2007.

[8] M. K. Jeya Kumar and R. S. Rajesh, "Performance analysis of MANET routing protocols in different mobility models," in *Proceedings of the International Journal of Computer Science and Network Security, IJCSNS*, vol. 9, no. 2, pp. 22-29, February 2009.