International Journal of Research (IJR) Vol-1, Issue-7, August 2014 **ISSN 2348-6848**

A Survey on TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) over AODV Routing Protocol

Robin Singh [1], Praveen Tripathi [2], Rahul Singh [3]
Department of Computer Science and Engineering
Kanpur Institute of Technology, Kanpur
singh.robin1988@gmail.com[1], praveentri81@gmail.com, rahulamreetsingh@gmail.com[3]

Abstract: *A mobile Ad-hoc network (MANET) is a collection of wireless nodes aimed at information exchange and resource sharing. The Ad hoc On-Demand Distance Vector (AODV) a well-known and widely used protocol for MANETs. This paper presents the study of TCP and UDP (two transport layer protocols) over AODV in a Mobile Ad hoc Network. The study is done on the TCP and UDP transport layer protocol and performance metrics like QoS defines a guarantee given by the network to satisfy a set of predetermined service performance constraints for the user in terms of end-to-end delay, jitter, and available bandwidth. Therefore, routing protocols must be feasible for all kinds of constrained applications to run well in a MANET. However, it is a significant technical challenge to define a comprehensive framework for QoS support, due to dynamic topology, distributed management and multi-hop connections for MANETs.*

Keywords: *TCP, UDP, AODV, MANET, QoS.*

INTRODUCTION

The Mobile Ad-hoc Network (MANET) is a collection of wireless nodes communicating over the wireless medium without any infrastructure. Working in ad-hoc mode allows all wireless devices within range, to discover and to communicate in peer-to-peer fashion without involving any base station. These nodes form a random topology, where the routers are free to move arbitrarily and arrange themselves as required. In MANET communication between two mobile nodes is made possible by providing a multi-hop path due to shorter range of radio signals. They offer transmission speeds of several Mbps, making possible the support of multimedia applications and real-time communications in MANETs. Since the area that must be covered may exceed the transmission range of the wireless devices, suitable routing protocols must be used to permit multi-hop communication, where as a hop is considered each wireless device. As the ad-hoc network has limited bandwidth, frequent topology changes and the energy limitations of the

mobile hosts, the protocols used for wired networks are considered inappropriate for MANETs. Therefore, for this highly dynamic environment new routing protocols have been designed. These protocols are classified as either proactive or reactive. The proactive (or table-driven) protocols maintain their routes through periodic updates and each node locally maintains routing tables. On the other hand, in the reactive (or on-demand) protocols the routes are established when required. The proactive approach provides fast response to route requests, but consumes more bandwidth since the network connectivity information must be continuously updated to reflect topology changes [1].

Many Researchers have gone through TCP and UDP for different routing protocols using various mobility models. The Random Waypoint mobility model is particularly popular for TCP as its performance may be affected by the basic initialization of other mobility models such as Reference Point Group mobility models (RPGM), Freeway model, Manhattan Model (MM) [2].

We have selected AODV as a routing protocol. This protocol is widely used in wireless environment and has numerous implementations available. The main focus of our paper is the comparative analysis of TCP and UDP over AODV protocol with respect to mobility, with varying pause time, node speed and node density. Our work participates to show that in which environment AODV works better for different QoS metrics.

RELATED WORK

The AODV routing protocol is developed in 1994 by C. Perkins. It uses a modified version of Bellman-Ford algorithm. For each node a sequence number is maintained which is generated from the destination, Bellman-Ford is usually used only when there are negative edge weights. The algorithm was developed by Richard Bellman and Lester Ford, Jr [3].

Authors discussed the performance and comparison of wireless routing protocols including DSDV [4]. They simulate fixed size network with varying pause time and velocity. They used simulation model with dynamic network size and pause time remained zero. The simulation measures all the QoS of the routing protocols, e.g. delay, jitter, throughput, loss ratio.

In [5], authors have discussed the performance evaluation of DSR, AODV and DSDV in grid environment to evaluate which routing protocol gives best performance in target mobile grid application. According to author, performance of DSR decreases with mobility with high rate as compare to DSDV and AODV.

The major issues of Ad hoc environment has discussed by authors in [6], it concern with energy consumption due to mobility. Mobile nodes are battery operated is a well known fact. OLSR and DSDV conserve less energy as compare to DSR and AODV.

The performance evaluation of, AODV, DSDV, DSR and TORA ad hoc routing protocols focusing on their suitability to support real time applications [1]. Simulations are done for a wide range of mobility and traffic scenarios.

MANET reactive and proactive routing protocols comparison evaluations has studied [7]. The protocols that use in MANET are DSDV, AODV, DSR, TORA, TORA, WRP ZRP and many more. The performance of these protocols are compared in term of packet delivery fraction, end-to-end delay normalized routing load routing overhead for 50 nodes and 100 nodes network model with different number of sources.

Most of the related work has done experiments on DSDV with TCP or UDP connection individually, but not on both comparatively. Our approach is novel on the basis of both TCP and UDP combine evaluation. All our effort is to identify in which environment TCP work well and UDP as well.

TRANSMISSION CONTROL PROTOCOL (TCP)

The Transmission Control Protocol (TCP) is one of the core protocols of the Internet protocol suite (IP), and is so common that the entire suite is often called TCP/IP. TCP provides reliable, ordered and error-checked delivery of a stream of octets between programs running on computers connected to a local area network, intranet or the public Internet. It resides at the transport layer.

Web browsers use TCP when they connect to servers on the World Wide Web, and it is used to deliver email and transfer files from one location to another. HTTP, HTTPS, SMTP, POP3, IMAP, SSH, FTP, Telnet and a variety of other protocols are typically encapsulated in TCP.

The protocol corresponds to the transport layer of TCP/IP suite. TCP provides a communication service at an intermediate level between an application program and the Internet Protocol (IP). That is, when an application program desires to send a large chunk of data across the Internet using IP, instead of breaking the data into IP-sized pieces and issuing a series of IP requests, the software can issue a single request to TCP and let TCP handle the IP details.

IP works by exchanging pieces of information called packets. A packet is a sequence of octets (bytes) and consists of a *header* followed by a *body*. The header describes the packet's source, destination and control information. The body contains the data IP is transmitting.

Due to network congestion, traffic load balancing, or other unpredictable network behavior, IP packets can be lost, duplicated, or delivered out. TCP detects these problems, requests retransmission of lost data, rearranges out-of-order data, and even helps minimize network congestion to reduce the occurrence of the other problems. Once the TCP receiver has reassembled the sequence of octets originally transmitted, it passes them to the receiving application. Thus, TCP abstracts the application's communication from the underlying networking details.

TCP is optimized for accurate delivery rather than timely delivery, and therefore, TCP sometimes incurs relatively long delays (on the order of seconds) while waiting for out-of-order messages or retransmissions of lost messages. It is not particularly suitable for real-time applications such as voice over IP. For such applications, protocols like the Real time transport Protocol (RTP) running over the User Datagram Protocol (UDP) are usually recommended instead.

TCP is a reliable stream delivery service that guarantees that all bytes received will be identical with bytes sent and in the correct order. Since packet transfer over many networks is not reliable, a technique known as positive acknowledgment with retransmission is used to guarantee reliability of packet transfers. This fundamental technique requires the receiver to respond with an acknowledgment message as it receives the data. The sender keeps a record of each packet it sends. The sender also maintains a timer from when the packet was sent, and retransmits a packet if the timer expires before the message has been acknowledged. The timer is needed in case a packet gets lost or corrupted.

While IP handles actual delivery of the data, TCP keeps track of the individual units of data transmission, called *segments* that a message is divided into for efficient routing through the network.

TCP protocol operations may be divided into three phases. Connections must be properly established in a multi-step handshake process (connection establishment) before entering the data transfer phase. After data transmission is completed, the connection termination closes established virtual circuits and releases all allocated resources.

USER DATAGRAM PROTOCOLS (UDP)

The User Datagram Protocol (UDP) is one of the core members of the Internet protocol suite. With UDP, computer applications can send messages, in this case referred to as *datagram's*; to other hosts on an Internet Protocol (IP) network without prior communications to set up special transmission channels or data paths. The protocol was designed by David P. Reed in 1980 and formally defined in RFC 768.

UDP uses a simple transmission model with a minimum of protocol mechanism. It has no handshaking dialogues, and thus exposes any unreliability of the underlying network protocol to the user's program. As this is normally IP over unreliable media, there is no guarantee of delivery, ordering, or duplicate protection. UDP provides checksums for data integrity, and port numbers for addressing different functions at the source and destination of the datagram.

UDP is suitable for purposes where error checking and correction is either not necessary or is performed in the application, avoiding the overhead of such processing at the network interface level. Time-sensitive applications often use UDP because dropping packets is preferable to waiting for delayed packets, which may not be an option in a real-time system.[2] If error correction facilities are needed at the network interface level, an application may use the Transmission Control Protocol (TCP) or Stream Control Transmission Protocol (SCTP) which are designed for this purpose.

A number of UDP's attributes make it especially suited for certain applications.

- It is transaction-oriented, suitable for simple query-response protocols such as the Domain Name System or the Network Time Protocol.
- It provides datagrams, suitable for modeling other protocols such as in IP tunneling or Remote Procedure Call and the Network File System.

- It is simple, suitable for bootstrapping or other purposes without a full protocol stack, such as the DHCP and Trivial File Transfer Protocol.
- It is stateless, suitable for very large numbers of clients, such as in streaming media applications for example IPTV.
- The lack of retransmission delays makes it suitable for real-time applications such as Voice over IP, online games, and many protocols built on top of the Real Time Streaming Protocol.
- Works well in unidirectional communication, suitable for broadcast information such as in many kinds of service discovery and shared information such as broadcast time or Routing Information Protocol.

COMPARISON OF TCP AND UDP

Transmission Control Protocol is a connection-oriented protocol, which means that it requires handshaking to set up end-to-end communications. Once a connection is set up, user data may be sent bi-directionally over the connection.

Reliable – TCP manages message acknowledgment, retransmission and timeout. Multiple attempts to deliver the message are made. If it gets lost along the way, the server will re-request the lost part. In TCP, there's either no missing data, or, in case of multiple timeouts, the connection is dropped.

Ordered – If two messages are sent over a connection in sequence, the first message will reach the receiving application first. When data segments arrive in the wrong order, TCP buffers delay the out-of-order data until all data can be properly re-ordered and delivered to the application.

Heavyweight – TCP requires three packets to set up a socket connection, before any user data can be sent. TCP handles reliability and congestion control.

Streaming – Data is read as a byte stream, no distinguishing indications are transmitted to signal message (segment) boundaries.

UDP is a simpler message-based connectionless protocol. Connectionless protocols do not set up a dedicated end-to-end connection. Communication is achieved by transmitting information in one direction from source to destination without verifying the readiness or state of the receiver. However, one primary benefit of UDP over TCP is the application to VoIP where latency and jitter are the primary concerns. It is assumed in VoIP UDP that the end users provide any necessary real time confirmation that the message has been received.

Unreliable – When a message is sent, it cannot be known if it will reach its destination; it could get lost along the way. There is no concept of acknowledgment, retransmission, or timeout.

Not ordered – If two messages are sent to the same recipient, the order in which they arrive cannot be predicted.

Lightweight – There is no ordering of messages, no tracking connections, etc. It is a small transport layer designed on top of IP.

Datagrams – Packets are sent individually and are checked for integrity only if they arrive. Packets have definite boundaries which are honored upon receipt, meaning a read operation at the receiver socket will yield an entire message as it was originally sent.

No congestion control – UDP itself does not avoid congestion, unless they implement congestion control measures at the application level.

AD-HOC ON-DEMAND DISTANCE VECTOR ROUTING (AODV)

In AODV, the network is silent until a connection is needed. At that point the network node that needs a connection broadcasts a request for connection. Other AODV nodes forward this message, and record the node that they heard it from, creating an explosion of temporary routes back to the needy node. When a node receives such a message and already has a route to the desired node, it sends a message backwards through a temporary route to the requesting node. The needy node then begins using the route that has the least number of hops through other nodes. Unused entries in the routing tables are recycled after a time.

When a link fails, a routing error is passed back to a transmitting node, and the process repeats. Much of the complexity of the protocol is to lower the number of messages to conserve the capacity of the network. For example, each request for a route has a sequence number. Nodes use this sequence number so that they do not repeat route requests that they have already passed on. Another such feature is that the route requests have a "time to live" number that limits how many times they can be retransmitted. Another such feature is that if a route request fails, another route request may not be sent until twice as much time has passed as the timeout of the previous route request.

The advantage of AODV is that it creates no extra traffic for communication along existing links. Also, distance vector routing is simple, and doesn't require much memory or calculation. However AODV requires more time to establish a connection, and the initial communication to establish a route is heavier than some other approaches.

The AODV Routing Protocol uses an on-demand approach for finding routes, that is, a route is established only when it is required by a source node for transmitting data packets. It employs destination sequence numbers to identify the most recent path. The major difference between AODV and Dynamic Source Routing (DSR) stems out from the fact that DSR uses source routing in which a data packet carries the complete path to be traversed. However, in AODV, the source node and the intermediate nodes store the next-hop information corresponding to each flow for data packet transmission. In an on-demand routing protocol, the source node floods the *RouteRequest* packet in the network when a route is not available for the desired destination. It may obtain multiple routes to different destinations from a single *RouteRequest.* The major difference between AODV and other on-demand routing protocols is that it uses a *destination sequence number* (DestSeqNum) to determine an up-to-date path to the destination. A node updates its path information only if the *DestSeqNum* of the

current packet received is greater or equal than the last *DestSeqNum* stored at the node with smaller hop count.

A RouteRequest carries the *source identifier* (SrcID), the *destination identifier* (DestID), the *source sequence number* (SrcSeqNum), the *destination sequence number* (DestSeqNum), the *broadcast identifier* (BcastID), and the *time to live* (TTL) field. DestSeqNum indicates the freshness of the route that is accepted by the source. When an intermediate node receives a RouteRequest, it either forwards it or prepares a RouteReply if it has a valid route to the destination. The validity of a route at the intermediate node is determined by comparing the sequence number at the intermediate node with the destination sequence number in the RouteRequest packet. If a RouteRequest is received multiple times, which is indicated by the BcastID-SrcID pair, the duplicate copies are discarded. All intermediate nodes having valid routes to the destination, or the destination node itself, are allowed to send RouteReply packets to the source. Every intermediate node, while forwarding a RouteRequest, enters the previous node address and its BcastID. A timer is used to delete this entry in case a RouteReply is not received before the timer expires. This helps in storing an active path at the intermediate node as AODV does not employ source routing of data packets. When a node receives a RouteReply packet, information about the previous node from which the packet was received is also stored in order to forward the data packet to this next node as the next hop toward the destination.

PERFORMANCE METRIC

QoS defines a guarantee given by the network to satisfy a set of predetermined service performance constraints for the user in terms of end-to-end delay, jitter, and available bandwidth.

Therefore, routing protocols must be feasible for all kinds of constrained applications to run well in a MANET. However, it is a significant technical challenge to define a comprehensive framework for QoS support, due to dynamic topology, distributed management and multi-hop connections for MANETs.

Throughput: It is one of the dimensional parameters of the network which gives the fraction of the channel capacity used for useful transmission selects a destination at the beginning of the simulation i.e., information whether or not data packets correctly delivered to the destinations.

Packet Delivery Ratio: Number of Data Packets Delivered over Number of Data Packets Generated. “Number of Data Packets Delivered” is the total number of received data packets by destinations; “Number of Data Packets Generated” is the total number of generated data packets by sources.

Average End to End Delay: average packet delivery time from a source to a destination. First for each source-destination pair, an average delay for packet delivery is computed. Then the whole average delay is computed from each pair average delay.

Drop Ratio: Packet Drop rate is one of the indicators for network congestion. In wireless environment, due to the physical

media and bandwidth limitations, the chance for packet dropping is increased. Therefore we choose it as one metric.

CONCLUSION

In this paper we have studied the AODV routing protocols over TCP and UDP Transport Layer. TCP is connection oriented protocol while UDP is connection less protocol. The study of the protocols was done with respect to metrics like Throughput, Packet delivery ratio, end to end delay and Drop Ratio.

REFERENCES

[1] Stavroulopoulos, T. Antonakopoulos and V. Makios, 2001. “Performance evaluation of mobile ad hoc network routing protocols for real time applications support,” in Proceedings of COMCON ‘8, Crete, Greece,

[2] Divecha, B., A. Abraham, C. Grosan and S. Sanyal, 2007. Impact of node mobility on MANET routing protocols models. Presented at JDIM, pp: 19-23.

[3] Perkins, E. and P. Bhagwat, 1994. “Highly dynamic Destination- Sequenced Distance-Vector routing (DSDV) for mobile computers,” in Proceedings of the SIGCOMM ’94 Conference on Communications Architectures, Protocols and Applications, pp: 234-244.

[4] Yuan, Peiyan and L.I. Layuan, 2006. Performance evaluation and simulations of routing protocols in ad hoc networks. In Proceedings of the 2006 workshop on Broadband wireless access for ubiquitous networking (BWAN '06). ACM, New York, NY, USA, Article 2. DOI= 10.1145/1189186.1189189 http:// doi.acm.org/ 10.1145/1189186.1189189.

[5] Usop, N.S.M., A. Abdullah and A.F.A. Abidin, 2009. Performance evaluation of aodv, dsdv and dsr routing protocol in grid environment. IJCSNS, 9(7): 261-268.

[6] Ouakil, L., S. Senouci, G. Pujolle and V.I. University de Paris, 2002. Performance comparison of ad hoc routing protocols based on energy consumption. In Ambience Workshop.

[7] Azad, S., A. Rahman and F. Anwar, 2007. A performance comparison of proactive and reactive routing protocols of mobile ad-hoc network (manet). Journal of Engineering and Applied Sci., 2(5): 891-896.

[8] The CMU Monarch Project, “Wireless and Mobility Extensions to ns-2”, www.monarch.cs.cmu.edu/cmu-ns.html.

[9] Rai, R. M., & Kumar, M. M. (2014). Implementation and Performance Evaluation of Energy Constraint AODV Routing. *International Journal of Research*, *1*(5), 368-374.