ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

Performance Analysis of Adhoc Network Routing Protocols

K.Mariappan[1], Dr.M.Karnan[2]

Assistant Professor, Dept. of CSE, Aringer Anna College of Engineering and Technology, Palani, Tamilnadu, India[1]

Professor and Principal, Aringer Anna College of Engineering and technology, Palani, Dindigul, India[2]

ABSTRACT: Routing in ad-hoc networks is nontrivial due to highly dynamic environment. In recent years several routing protocols targeted at mobile ad-hoc networks are being proposed and prominent among them are DSDV, AODV, TORA, and DSR. This paper does the comprehensive analysis of routing protocols using ns2 simulator. All protocols are provided with identical traffic load and mobility patterns. We have considered TCP as transport protocol and FTP as traffic generator. Results indicate that the performance of proactive routing protocol DSDV is far better than remaining protocols for TCP based traffic. DSR which uses source routing is the best among reactive routing protocols. The analysis is significant because we considered all the metrics as suggested by RFC 2501 and till to-date there are a few comparisons based on TCP.

KEYWORDS: Destination Sequenced Distance Vector (DSDV); Dynamic Source Routing (DSR); Ad -hoc On-demand Distance Vector (AODV); The Temporally-Ordered Routing Algorithm (TORA);File transfer protocol (FTP);transmission control protocol(TCP)

I. INTRODUCTION

An ad-hoc network is a collection of nodes forming a temporary network without the aid of any additional infrastructure and no centralized control. The nodes in an ad-hoc network [1] can be a laptop, PDA, or any other device capable of transmitting and receiving information. Nodes act both as an end system (transmitting and receiving data) and as a router (allowing traffic to pass through) resulting in multihop routing. Network is temporary as nodes are generally mobile and may go out of range of other nodes in the network. Routing in an ad-hoc network is nontrivial as they posses few characteristics [2] which make them different from wired networks. They are as follows:

- High probability of errors due to various transmission impairments
- Low Transmission range to conserve energy
- Frequent link breakages due to mobility
- Sleep period of operation of nodes and unidirectional links
- Unfavourable environmental conditions by virtue of applications of ad-hoc networks
- Looping problem due to mobility
- No proper Addressing scheme

Routing in ad-hoc networks [3] started with the two most successful routing algorithms of wired networks: Distance Vector and Link State. Compared to Link state method, Distance vector is computationally more efficient, easier to implement and requires much less storage space. However, it is well known that this algorithm can cause the formation of both short-lived and long-lived loops (Count-to-Infinity). Almost all proposed modifications to this algorithm eliminate the looping problem by forcing all nodes in the network to participate in some form of inter nodal coordination protocol. Such inter nodal coordination mechanisms might be effective when topological changes are rare. However, within an ad-hoc mobile environment enforcing any such inter nodal coordination mechanism will be difficult due to rapidly changing topology. Furthermore, the techniques split horizon and poisoned reverse are not useful within the wireless environment due to the broadcast nature of the transmission medium.

Link state algorithms are free of Count-to-infinity problem. However, they need to maintain the up-to-date version of the entire network topology at every node, which may constitute excessive storage and communication overhead in a

DOI: 10.15680/IJIRCCE.2016. 0407204

highly dynamic network. Besides, Link-state algorithms proposed or implemented to-date does not eliminate the creation of temporary routing loops. Some of the link costs in a node's view can be incorrect because of long propagation delays, partitioned network, etc. Such inconsistent views of network topologies might lead to formation of routing loops. These loops, however, are short lived because they disappear in the time it takes a message to traverse the diameter of the network. Wired networks are usually explicitly configured to have a link connecting two nodes, but there are no explicit links in ad-hoc network, and all communication is by broadcast transmission. The redundant paths in a wireless environment unnecessarily increase the size of routing updates that must be sent over the network, and increase the CPU overhead required to process each update and to compute new routes.

This paper is not first to provide a quantitative analysis of routing protocols for ad-hoc networks. But all these papers [4] [5] [6] [7] [8] did the comparison considering only few characteristics that should be possessed by routing protocols and all consider UDP as transport protocol. This paper is comprehensive study and considers all the characteristics as suggested by RFC 2501.

II. ROUTING PROTOCOLS OVERVIEW

A. DSDV

Destination Sequenced Distance Vector (DSDV) [9] is a Proactive routing protocol that solves the major problem associated with the Distance Vector routing of wired networks i.e., Count-to-infinity, by using Destination sequence numbers. Destination sequence number is the sequence number as originally stamped by the destination. The DSDV protocol requires each mobile station to advertise, to each of its current neighbours, its own routing table (for instance, by broadcasting its entries). The entries in this list may change fairly dynamically over time, so the advertisement must be made often enough to ensure that every mobile computer can almost always locate every other mobile computer. In addition, each mobile computer agrees to relay data packets to other computers upon request. At all instants, the DSDV protocol guarantees loop-free paths to each destination. Routes with more recent sequence numbers are always preferred as the basis for making forwarding decisions, but not necessarily advertised. Of the paths with the same sequence number, those with the smallest metric will be used.

The routing updates are sent in two ways: a "full dump" or incremental update. A full dump sends the full routing table to the neighbours and could span many packets whereas, in an incremental update only those entries from the routing table are sent that has a metric change since the last update and it must fit in a packet. When the network is relatively stable, incremental updates are sent to avoid extra traffic and full dump are relatively infrequent. In a fast changing network, incremental packets can grow big, so full dumps will be more frequent.

The updates can be time triggered (periodic) or event triggered. When any new or substantially modified route information is received by a Mobile Host, the new information will be retransmitted soon (subject to constraints imposed for damping route fluctuations). When a stabilized route shows a different metric for some destination that would likely constitute a significant change that needed to be advertised after stabilization. If a new sequence number for a route is received, but the metric stays the same, that would be unlikely to be considered as a significant change. Newly recorded routes are scheduled for immediate advertisement to the current Mobile Host's neighbours. Routes which show an improved metric are scheduled for advertisement at a time which depends on the average settling time for routes to the particular destination under consideration. A broken link is described by a metric of infinity (i.e., any value greater than the maximum allowed metric). When a link to a next hop has broken, any route through that next hop is immediately assigned infinity metric and assigned an updated sequence number. Since this qualifies as a substantial route change, such modified routes are immediately disclosed in a broadcast routing information packet.

B. DSR

Dynamic Source Routing (DSR) [10] is a reactive protocol i.e. it doesn't use periodic advertisements. It computes the routes when necessary and then maintains them. Source routing is a routing technique in which the sender of a packet determines the complete sequence of nodes through which the packet has to pass; the sender explicitly lists this route in the packet's header, identifying each forwarding "hop" by the address of the next node to which to transmit the packet on its way to the destination host. There are two significant stages in working of DSR: Route Discovery and Route Maintenance. A host initiating a route discovery broadcasts a route request packet which may be received by those hosts within wireless transmission range of it. The route request packet identifies the host, referred to as the target of the route discovery, for which the route is requested. If the route discovery is successful the initiating host receives a route reply packet listing a sequence of network hops through which it may reach the target. In addition to the address of the original initiator of the request and the target of the request, each route request packet contains a route record, in which is accumulated a record of the sequence of hops taken by the route request packet as it is propagated through the network during this route discovery.

While a host is using any source route, it monitors the continued correct operation of that route. This monitoring of the correct operation of a route in use is called route maintenance. When route maintenance detects a problem with a route in use, route discovery may be used again to discover a new, correct route to the destination. To optimize route discovery process, DSR uses cache memory efficiently. Suppose a host receives a route request packet for which it is not the target and is not already listed in the route record in the packet, and for which the pair (initiator address, request id) is not found in its list of recently seen requests; if the host has a route cache entry for the target of the request, it may append this cached route to the accumulated route record in the packet, and may return this route in a route reply packet to the initiator without propagating (re-broadcasting) the route request. The delay for route discovery and the total number of packets transmitted can be reduced by allowing data to be piggybacked on route request packets.

DSR uses no periodic routing advertisement messages, thereby reducing network bandwidth overhead, particularly during periods when little or no significant host movement is taking place. DSR has a unique advantage by virtue of source routing. As the route is part of the packet itself, routing loops, either short-lived or long-lived, cannot be formed as they can be immediately detected and eliminated.

C. AODV

Ad-hoc On-demand Distance Vector (AODV) [11] is essentially a combination of both DSR and DSDV. It borrows the basic on-demand mechanism of Route Discovery and Route Maintenance from DSR, plus the use of hop-by-hop routing, sequence numbers, and periodic beacons from DSDV. It uses destination sequence numbers to ensure loop freedom at all times and by avoiding the Bellman-Ford ”count-to-infinity” problem offers quick convergence when the ad hoc network topology changes Route Requests (RREQs), Route Replies (RREPs), and Route Errors (RERRs) are the message types defined by AODV. These message types are received via UDP, and normal IP header processing applies.

As long as the endpoints of a communication connection have valid routes to each other, AODV does not play any role. When a route to a new destination is needed, the node broadcasts a RREQ to find a route to the destination. A route can be determined when the RREQ reaches either the destination itself, or an intermediate node with a 'fresh enough' route to the destination. A 'fresh enough' route is a valid route entry for the destination whose associated sequence number is at least as great as that contained in the RREQ. The route is made available by unicasting a RREP back to the origination of the RREQ. Each node receiving the request caches a route back to the originator of the request, so that the RREP can be unicast from the destination along a path to that originator, or likewise from any intermediate node that is able to satisfy the request.

If intermediate nodes reply to every transmission of a given RREQ, the destination does not receive any copies of it. In this situation, the destination does not learn of a route to the originating node. This could cause the destination to initiate a route discovery (for example, if the originator is attempting to establish a TCP session). In order that the destinations learn of routes to the originating node, the originating node should set the “gratuitous RREP” ('G') flag in the RREQ if for any reason the destination is likely to need a route to the originating node. If in response to a RREQ with the 'G' flag set, an intermediate node returns a RREP, it MUST also unicast a gratuitous RREP to the destination node.

Nodes monitor the link status of next hops in active routes. In order to maintain routes, AODV normally requires that each node periodically transmit a HELLO message, with a default rate of once per every second. Failure to receive three consecutive HELLO messages from a neighbour is taken as an indication that the link to the neighbour in question is down. When a link break in an active route is detected, a RERR message is used to notify other nodes that the loss of that link has occurred. The RERR message indicates those destinations which are now unreachable due to the loss of the link. In order to enable this reporting mechanism, each node keeps a “precursor list”, containing the IP address for each of its neighbours that are likely to use it as a next hop towards the destination that is now unreachable.

D. TORA

The Temporally-Ordered Routing Algorithm (TORA) [12] is an adaptive routing protocol for multihop networks that possesses the following attributes:

- Distributed execution
- Multipath routing
- The protocol can simultaneously support both source-initiated, on-demand routing for some destinations and destination-initiated, proactive routing for other destinations.
- Minimization of communication overhead via localization of algorithmic reaction to topological changes.

TORA is distributed, in that routers need only maintain information about adjacent routers (i.e., one-hop knowledge). Like a distance-vector routing approach, TORA maintains state on a per-destination basis. However, TORA does not continuously execute a shortest-path computation and thus the metric used to establish the routing structure does not represent a distance. The destination-oriented nature of the routing structure in TORA supports a mix of reactive and proactive routing on a per-destination basis. During reactive operation, sources initiate the establishment of routes to a given destination on-demand. This mode of operation may be advantageous in dynamic networks with relatively sparse traffic patterns, since it may not be necessary (or desirable) to maintain routes between every source/destination pair at all times. At the same time, selected destinations can initiate proactive operation, resembling traditional table-driven routing approaches. This allows routes to be proactively maintained to destinations for which routing is consistently or frequently required (e.g., servers or gateways to hardwired infrastructure). TORA is designed to minimize the communication overhead associated with adapting to network topological changes. The scope of TORA's control messaging is typically localized to a very small set of nodes near a topological change.

The design and flexibility of TORA allow its operation to be biased towards high reactivity (i.e., low time complexity) and bandwidth conservation (i.e., low communication complexity) rather than routing optimality--making it potentially well-suited for use in dynamic wireless networks.

A logically separate version of TORA is run for each "destination" to which routing is required. TORA assigns directions to the links between routers to form a routing structure that is used to forward datagram’s to the destination. A router assigns a direction ("upstream" or "downstream") to the link with a neighbouring router based on the relative values of a metric associated with each router. The metric maintained by a router can conceptually be thought of as the router's "height" (i.e., links are directed from the higher router to the lower router). The significance of the heights and the link directional assignments is that a router may only forward datagram’s downstream. Links from a router to any neighbouring routers with an unknown or undefined height are considered undirected and cannot be used for forwarding. Collectively, the heights of the routers and the link directional assignments form a loop-free, multipath routing structure in which all directed paths lead downstream to the destination.

The TORA link reversal process creates short-lived routing loops that exist from the time that the link-reversal starts until the time that all nodes that need to be aware of the reversal receive the corresponding update. Delaying the transmission of TORA routing messages for aggregation, coupled with any queuing delay at the network interface, allows these routing loops to last long enough that significant numbers of data packets are dropped.

III.SIMULATION

Our protocol evaluations are based on the simulation using ns2 [13] and the graphs are generated using X-graph. NS2 is a discrete event simulator developed by the University of California at Berkeley and the VINT project. NS2 supports two languages, system programming language C++ for detail implementation and scripting language TCL for configuring and experimenting with the different parameters quickly. NS2 has all the essential features like abstraction, visualization, emulation, and traffic & scenario generation. X-graph draws a graph on a display with data given either from data files or standard input. It can display up to 64 independent data sets using different colours and line styles for each set.

Simulation environment consists of 50 wireless nodes forming an ad hoc network, moving about over a 670 X 670 flat space for 200 seconds of simulated time. The physical radio characteristics of each mobile node’s network interface, such as the antenna gain, transmit power, and receiver sensitivity, were chosen to approximate the Lucent Waveland direct sequence spread spectrum radio. In order to enable direct, fair comparisons between the protocols, it was critical to challenge the protocols with identical loads and environmental conditions. Each run of the simulator accepts as input a scenario file that describes the exact motion of each node and the exact sequence of packets originated by each node, together with the exact time at which each change in motion or packet origination is to occur. We pre-generated 45 different scenario files with varying movement patterns and traffic loads (FTP), and then ran all four routing protocols against each of these scenario files. Since each protocol was challenged in an identical fashion, we can directly compare the performance results of the four protocols.

A. *Movement Model*

Nodes in the simulation move according to a model that we call the “random waypoint” model. The movement scenario files we used for each simulation are characterized by a pause time. Each node begins the simulation by remaining stationary for pause time seconds. It then selects a random destination in the 670 X 670m space and moves to that destination at a speed distributed uniformly between 0 and some maximum speed. Upon reaching the destination, the node pauses again for pause time seconds, selects another destination, and proceeds there as previously described, repeating this behaviour for the duration of the simulation. Each simulation ran for 200 seconds of simulated time. We ran our simulations with movement patterns generated for 9 different pause times: 0, 5, 10, 20, 30, 50, 100, 150, 200 seconds and 5 different speeds: 2, 5, 10, 15, and 20. A pause time of 0 seconds corresponds to continuous motion, and a pause time of 200 (the length of the simulation) corresponds to no motion.

B. *Metrics*

In comparing the protocols, we chose to evaluate them according to the following metrics:

Throughput: It is defined as total number of packets received by the destination. It is a measure of effectiveness of a routing protocol. Finally what matters is the number of packets delivered successfully.

Packet delivery ratio: the ratio between the number of packets received by the TCP sink at the final destination and the number of packets originated by the “application layer” sources. It is a measure of efficiency of the protocol.

Routing overhead: The total number of routing packets transmitted during the simulation. For packets sent over multiple hops, each transmission of the packet (each hop) counts as one transmission. Since End-to-end Network Throughput (data routing performance) is defined as the external measure of effectiveness, efficiency is considered to be the internal measure. To achieve a given level of data routing performance, two different protocols can use differing amounts of overhead, depending on their internal efficiency, and thus protocol efficiency may or may not directly affect data routing performance. If control and data traffic share the same channel, and the channels capacity is limited, then excessive control traffic often impacts data routing performance.

Path optimality: The difference between the number of hops a packet took to reach its destination and the length of the shortest path that physically existed through the network when the packet was originated.

Packets lost: it is a measure of the number of packets dropped by the routers due to various reasons. The reasons we have considered for evaluation are Collisions, time outs, looping, errors.

Average Delay: It is a metric which is very significant with multimedia and real-time traffic. It is very important for any application where data is processed online.

IV. SIMULATION RESULTS

The simulation results bring out some important characteristic differences between the routing protocols.

Though proactive protocols are intrinsically not suitable for any mobile network because they involve periodic exchange of information resulting in consumption of energy of battery operated nodes; they are capable of maintaining a connection which is required for TCP traffic. From the Fig.1 it can be inferred that DSDV throughput is far better than other protocols. Since DSR pre-computes the routes before sending the packets its packet delivery ratio is better than other protocols as shown in Fig. 2. TORA’s performance is relatively poor when throughput and packet delivery ratio are considered as metrics

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

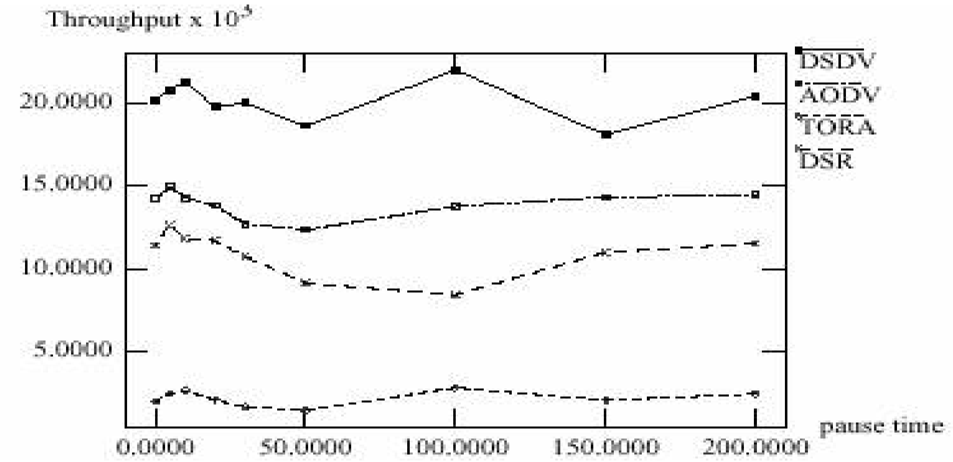


Figure 1: Total number of packets received at various levels of mobility

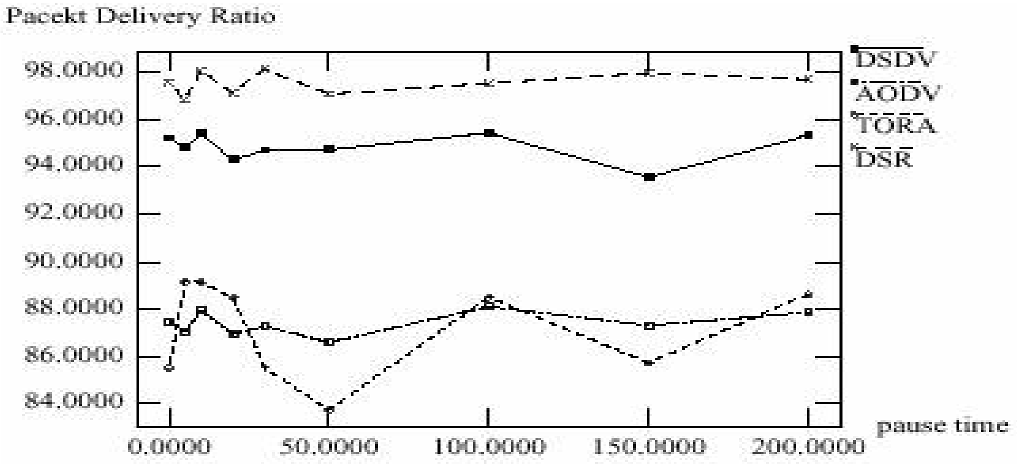


Figure 2: Ratio of packets delivered/Packets transmitted at different levels of mobility

Since DSDV is a proactive routing protocol in most of the cases it uses already established route and tries to get rid of the packets immediately resulting in low average delay. DSR requires complete route at the source itself before transferring the packet and since it is reactive routing protocol significant delay is introduced before transferring the packet. AODV also introduces low delays when compared to DSR and TORA. All this can be inferred from Fig. 3. Routing overhead of all the protocols DSDV, AODV and DSR is significantly low as indicated in the Fig. 4

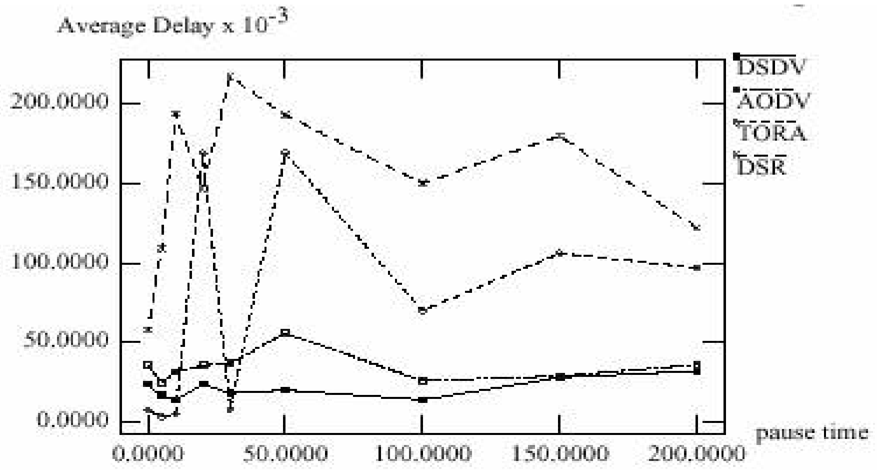


Figure 3: Delay introduced by routing protocols with variation in mobility

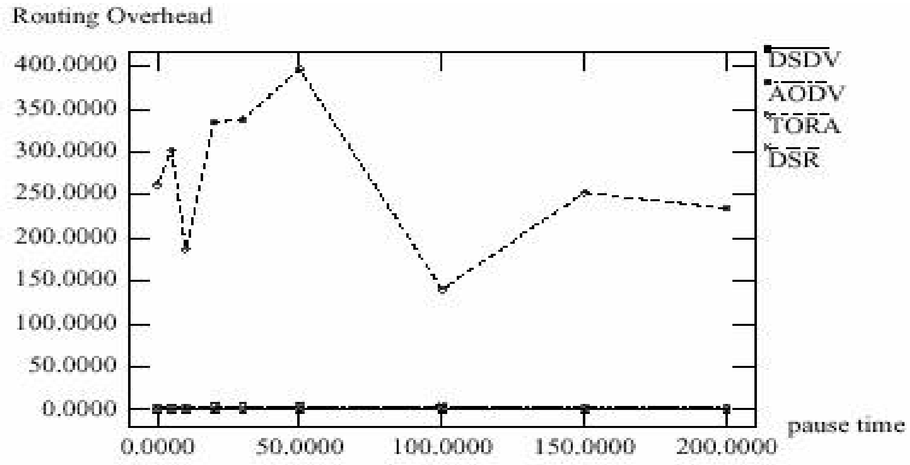


Figure 4: Variation of routing overhead with mobility

TORA drops few packets but its throughput is very low. DSDV is better than other two protocols in dropping packets as indicated in Fig. 5. DSDV uses the optimal path in more than 90% of the cases and the difference between optimal path and the forwarded path is negligible. Other protocols fail to use the optimal path and the difference between optimal path and forwarded path is more than 3 hops which can be determined from Fig. 6. The impact of mobility on all the protocols is surprisingly not significant.

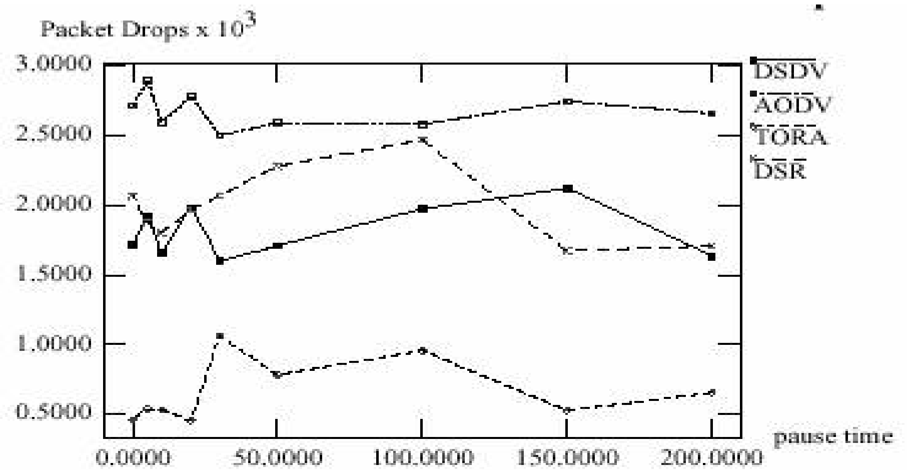


Figure 5: Total number of packets dropped with variation in mobility

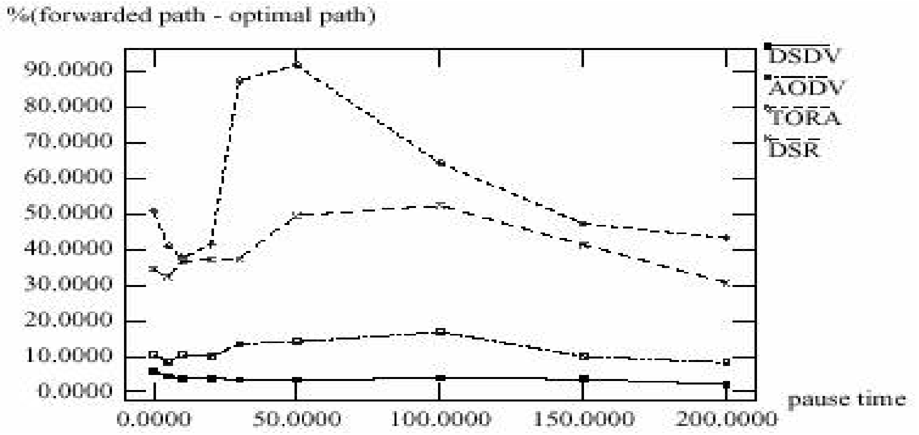


Figure 6: Percentage difference between forwarded path and optimal path with mobility

The variation of performance of all the protocols with speed is similar to the variation in performance with mobility as shown in Fig. 7-12. DSDV performs better than all the remaining protocols. The impact of speed on DSR is significant. Similar is the case with TORA

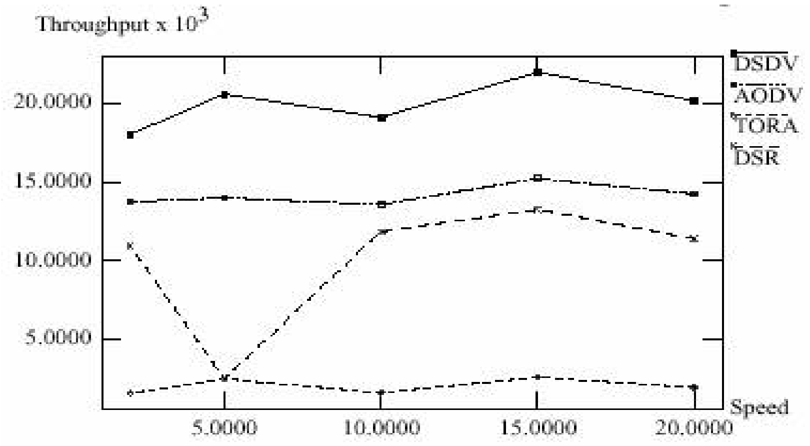


Figure 7: Total number of packets received at various levels of speed

DOI: 10.15680/IJIRCCE.2016. 0407204

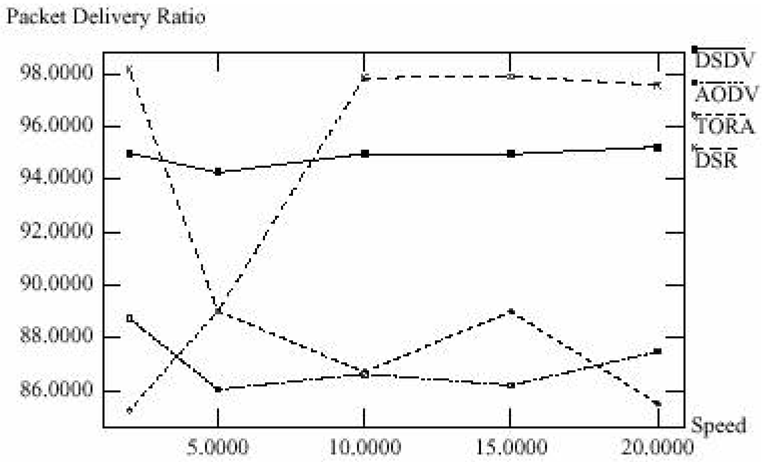


Figure 8: Ratio of packets delivered/Packets transmitted at different speeds

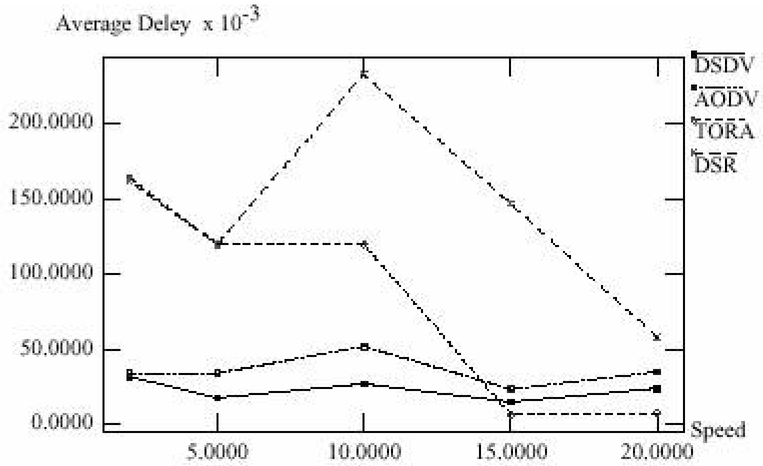


Figure 9: Delay introduced by routing protocols with variation in speed

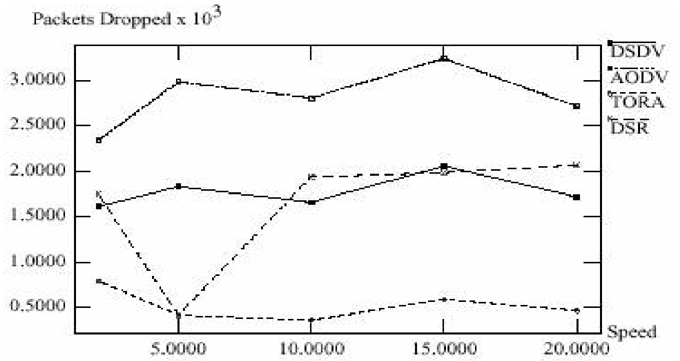
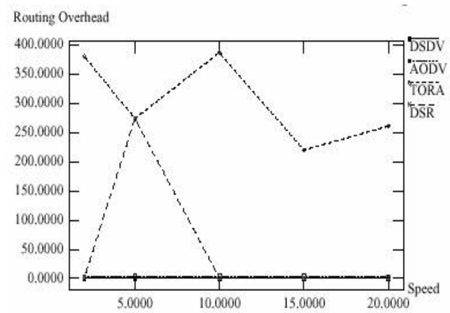


Figure 10: Total number of packets dropped with variation in speed



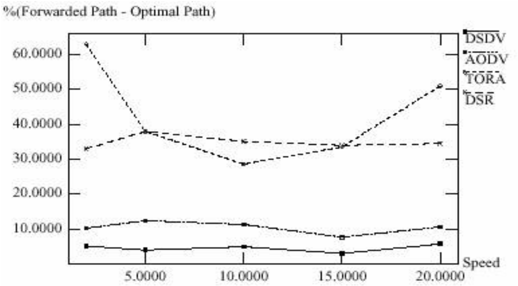


Fig11: Variation of Routing overhead with speed

Fig12: % difference between forwarded and optimal Path

V. CONCLUSION

This paper does the realistic comparison of four routing protocols DSDV, AODV, TORA and DSR. The significant observation is, simulation results agree with expected results based on theoretical analysis. As expected, proactive routing protocol DSDV performance is best considering its ability to maintain connection by periodic exchange of information, which is required for TCP, based traffic. Results are only valid when we consider TCP traffic and TCP is not appropriate transport protocol for highly mobile multihop networks and UDP is preferred. For UDP traffic performance of reactive routing protocols is better than proactive routing protocols.

REFERENCES

[1] J. Macker and S. Corson, Mobile Ad hoc Networks (MANET), http://www.ietf.org/charters/manet-charter.html, IETF Working Group Charter, 1997.

[2]. S. R. Das, C. Perkins, and E. Royer, Performance comparison of Two On-demand Routing Protocols for Ad Hoc Networks, Proc. of IEEE INFOCOM 2000, March 2000.

[3] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, and J. Jetcheva, A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols, Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), October 1998.

[4] Vincent D. Park and M. Scott Corson, A performance comparison of TORA and Ideal Link State routing, Proc. of IEEE Symposium on Computers and Communication, June 1998.

[5] Bernd Freisleben and Ralph Jansen, Analysis of routing protocols for ad hoc networks of mobile computers, Proc. of the 15th IASTED International Conference on Applied Informatics, pages 133–136, Innsbruck, Austria, February 1997.

[7] S. Corson, and J. Macker, Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, RFC 2501, January 1999.

[6] S.R. Das, R. Castaneda, J. Yan, and R. Sengupta, Comparative performance evaluation of routing protocols for mobile, ad hoc networks, Proc. of 7th International Conference on Computer Communications and Networks (IC3N), pages 153–161, October 1998.

[8] David B. Johnson, Routing in Ad Hoc Networks of Mobile Hosts, Proc. of the IEEE Workshop on Mobile Computing Systems and Applications, pages 158-163, December 1994.

[9] Charles E. Perkins and Pravin Bhagwat, Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers, Proc. of the SIGCOMM ’94 Conference on Communications Architectures, Protocols and Applications, pages 234–244, August 1994.

[10] David B. Johnson, David A. Maltz, and Yih-Chun Hu, The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR), <draft-ietf-manet-dsr-10.txt> Internet-draft, 19 July 2004.

[11] C. Perkins, E. Belding-Royer, and S. Das, Ad hoc On-Demand Distance Vector (AODV) Routing, RFC 3561, July 2003.

[12] V. Park and S. Corson, Temporally Ordered Routing Algorithm (TORA) Version 1, Functional specification IETF Internet draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-tora-spec-01.txt, 1998.

[13]. Kevin Fall and Kannan Varadhan, editors, NS-Documentation, http://www.isi.edu/nsnam/ns/ns-documentation.html, April 2005.