Ad-Hoc Data Processing and Its Relation with Cloud Computing Process Using Functional Approach

[1]M Dhasaratham,[2]Dr. R P Singh
[1]Research Scholar ,[2] Research Guide
[1]SSSUTMS, Sehore,India, [2]SSSUTMS, Sehore,India

***Abstract*—In this paper discussed about the main observation from the application oriented metrics such as delay and packet delivery fraction, DSR outperforms AODV we compare the performance of two on-demand routing protocols for mobile ad hoc networks – Dynamic Source Routing (DSR) and Ad Hoc On-Demand Distance Vector (AODV). The performance differentials are analyzed using varying traffic. The performance is evaluated by using Network Simulator, from viewpoint of packet delivery fraction and average end- to-end delay. A cloud algorithm is used to design a network and then complexity of the network is estimated. After that the lower and upper bounds are evaluated. By using functional approach, the computation load over the network can be distributed accordingly.**

***IndexTerms*— Cloud Computing, Ad hoc Network, Packet Delivery Fraction, Dynamic Source Routing**

I. INTRODUCTION

Mobile Ad hoc networks are forming their network without the help of any centralized administration at the time of moving dynamically. It's an infrastructure-less network comprising of mobile nodes which dynamically form a network. So here, the efficient routing protocols are needed for better performance in such networks. And then we compare the performance of two on-demand routing protocols for mobile ad hoc networks – **Dynamic Source Routing (DSR) and Ad Hoc On-Demand Distance Vector (AODV)**. The performance differentials are analyzed using varying traffic. The performance is evaluated by using Network Simulator (NS- 2.1), from viewpoint of packet delivery fraction and average end- to-end delay.

MANET consists of a set of wireless mobile nodes communicating with each other without any centralized control or fixed network infrastructure. MANET nodes are equipped with wireless transmitters and receivers. In MANET, nodes have to been announcing their presence periodically and listen for their neighbors announcements broadcast to discover and learn how to reach each other. Therefore, the new node is not familiar with its network topology at the beginning. Mobility and scalability are the main challenges in this kind of infrastructure- less networks. Where mobility means that often changes the network topology drastically and unpredictably while scalability means more traffic and overhead control packets. In addition, wireless links have been significantly affected by constraint resources such as bandwidth and power. Hence, there is a need of efficient and effective MANET routing protocols to allow the nodes to communicate over multi- hop paths, and should perform acceptably in a dynamic, low bandwidth environment.

The key motivation behind the design of on- demand protocols is the reduction of the routing load. High routing load usually has a significant performance impact on low bandwidth wireless links. On the other hand, the on – demand approach do not compute until there is a data to be sent. It may cause longer delay to construct a route, and the connection may not be set up due to the long latency and congested channel. So here, the performance comparison metrics are Packet delivery fraction and average End – to – End delay for the varying traffic sources are handled.

II.PROTOCOL DESCRIPTION

DSR

The main feature of the DSR is to use source routing. That is, the sender of a packet determines complete route from itself to the destination, including all intermediate hops. These routes are stored in route cache. The data packet carries the source route in the packet header . All intermediate hosts forward the packet based on this predetermined route (called source route). No routing decision is made at intermediate hosts. When a node in the ad hoc network attempts to send a data packet to a destination for which it does not already know the route, it uses the route discovery process to dynamically determine such a route. Route discovery process is straight forward; the initiator flooded the route request (RREQ) packet across the network. A RREQ packet contains the address of destination host as well as a route record which records the host that the request has passed.

Each node receiving a RREQ, rebroadcast it, unless it is the destination or it has a route to the destination in its route cache. In both cases, the complete route from the initiator to the destination is found. This route is then replies to the initiator by sending back the route reply (RREP) packet. The route request builds up path traversed so far. The path carried back by the RREP packet is cached at the source for further use. If any link on a source route is broken, the source node is notified using a route error (RERR) packet. The source removes any route using this link from its cache. A new route discovery process must be initiated by the source, if this route is still needed.

AODV

Ad Hoc On- Demand Distance Vector (AODV) combines the features of Destination Sequenced Distance Vector (DSDV) and Dynamic Source Routing (DSR) protocols. AODV uses destination sequence number like DSDV to determine the freshness of routing information. However, AODV maintains route in a distribution fashion, as routing table entries. The AODV keeps routing table entries in the form of <destination, next hop, distance>. An important feature of the AODV is maintenance of timer – based states in each node, regarding utilization of individual routing table entries. A routing table entry expires if not used recently. In AODV routing, when a source has data to transmit to a new destination, it broadcast a route request (RREQ) for that destination to its neighbors. RREQ packet contains the address of destination node, sequence number of destination node, broadcasting sequence number, sequence number of source node, address of previous hop count. When an intermediate node receives RREQ information, it would forward route reply (RREP) packet by the reverse routing. RREP containing address of source node, address of destination node, hop count and life time. The RREP was unicasted in hop – by – hop fashion to the source

When the source receives the RREP, it record route to the destination and begins sending data. If multiple route replies are received by the source, the route with the shortest hop count and highest destination number is chosen. In case link break is detected the node at the upstream of route broken would broadcast RERR, which contains the address and sequence number of unreachable nodes to the neighbor nodes. As the route error propagates towards the source, each intermediate node invalidates route to unreachable destinations. When the source node receives the RERR, it invalidates the route and reinitiate route discovery. Sequence number in AODV plays a key role in ensuring loop freedom and freshness of the route. A higher sequence number signifies a fresher route. AODV can handle low, moderate and relatively high mobility rates, as well as variety of data traffic.

III.PERFORMANCE ANALYZING AND SIMULATION SCENARIO

The performance comparison of the routing protocols was carried out based on two things they are Packet Delivery Fraction (PDF) and Average End-To-End Delay (EED) metrics. ***Packet delivery fraction (PDF)*** is the ratio of data packets delivered to the destination to those generated by CBR sources. This metric actually tells us how much reliable the protocol is. It describes the loss rate that will be seen by the transport protocol, which in turn affects, the maximum throughput of the ad hoc network.

$$PDF = \frac{\sum \text{CBR Received Packets by CBR destination}}{\sum \text{CBR Sent Packets by CBR Sources}} x100$$

Average end- to- end delay (EED) includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC propagation and transfer time. The Average EED is computed as follows.

$$EED = \frac{1}{N}\sum_{n=1}^{N}(r_n - s_n)$$

s_n=Time that data packet n was sent
r_n=Time that data packet n was received
N = Total number of data packets received

A simulated network scenario for variety of traffic loads were implemented on NS-2.1. The simulation parameters and their values are given in the following table,

Parameter	Value
Node density	50
Area	500mx500m
Simulation Time	900 sec.
Mobility model	Random Waypoint
Pause Time	50,100,150,220,325,575,800
Node speed	0-5m/s
Wireless MAC	802.11
Protocol used	DSR, AODV
Traffic Source	CBR, 4packet/sec, 512 byte
Number of Sources	10, 18, 32, 45

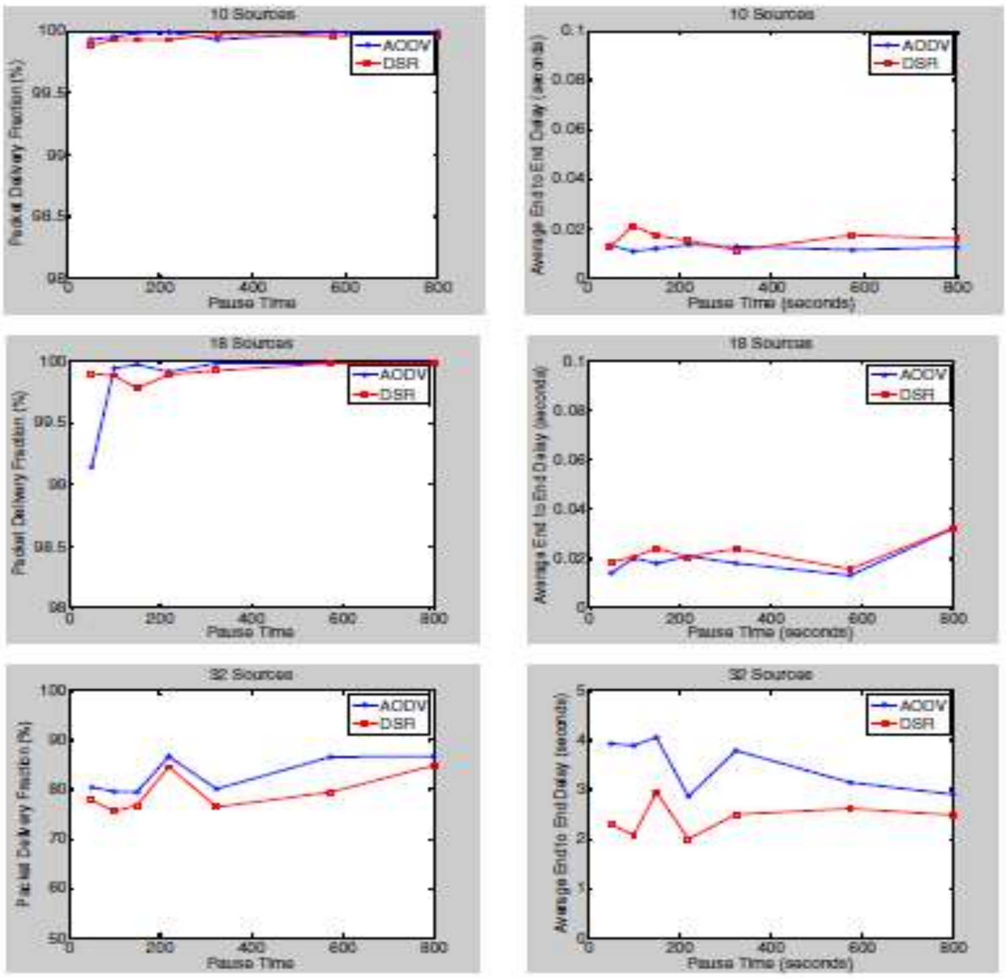
Table 1 Simulation parameters

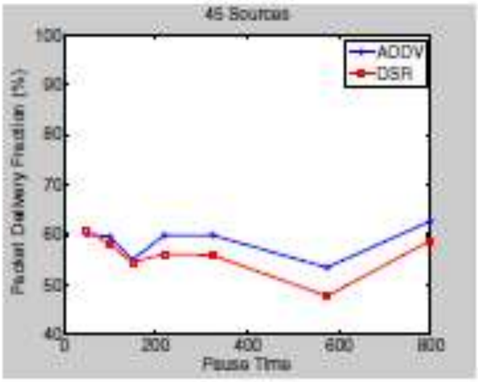
The 50 node density is taken in the square area of 500 m x 500 m. The simulation was carried out by the use of varying number of sources (10, 18, 32 and 45) with the moderate packet rate and changing pause time. The simulation is run for 900 seconds.

IV.RESULT AND DISCUSSION

The Figure 1 shows that the packet delivery fractions for DSR and AODV are very similar for 10 and 18 sources. The DSR outperforms AODV for high pause time at low traffic load as 10, 18. Since number of connections and mobility of nodes are low, and the entire route to the destination is available in DSR cache. Hence the packets were delivered to the destination faster than AODV. In contrast AODV does not have entire route to the destination and utilizes hop-by hop route. But when the sources become more, DSR with small pause time which means high mobility began to perform worse than AODV. Since AODV have more routing control packets, but may always choose the fresh route, while DSR with smaller number of routing packets under stressful situation with the network topology continue to change from time to time, will be inclined to choose wrong routes, thus lower the packet delivery rate.

Figure 2 shows that with the low traffic of 18 sources AODV have more delay than DSR, because AODV has much more routing packets than DSR, and those routing packets will consume more bandwidth. When load become heavy which is 32 sources, DSR with small pause time have more delay time since stale routes often be choose, which lost many delivery time. The delay for both the protocols increases with 45 sources at low mobility. This is due to the high level of network congestion and multiple access interference at certain regions of the ad hoc network. This phenomenon is less visible with higher mobility where traffic automatically gets more evenly distributed due to source movements.





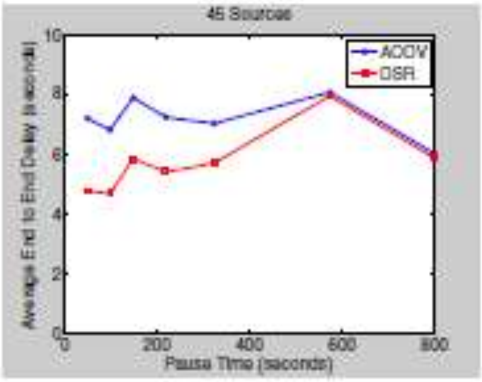


Figure 1 packet delivery fraction for the 50 Nodes model for varying no. of sources

Figure 2 Average End to End delay for the 50 nodes model for varying no. of sources

V.Conclusion

The main observation from the above simulation is that for application oriented metrics such as delay and packet delivery fraction, DSR outperforms AODV in less 'stressful' situations, i.e., smaller number of nodes and lower load. AODV however outperforms DSR in more stressful situations. The poor delay and PDF performances of DSR are mainly attributed to aggressive use of caching and lack of any mechanism to expire stale routes. Aggressive caching, however seems to help DSR at low loads and also keeps its routing loads down. The mechanism to expire routes and/or determine freshness of routes will benefit DSR's performance significantly. On the other hand AODV keeps track of actively used routes, destination also can be searched using a single route discovery flood to control routing load, which is totally timer –based.

REFERENCES

[1] Fantechi, S. Gnesi, G. Ristori, M. Carenini, M. Vanocchi, and P. Moreschini, Assisting Requirement Formalization by Means of Natural Language Translation, Formal Methods in System Design, 2012, 4(3):243-263.

[2] Fuxman, L. Liu, J. Mylopoulos, M. Roveri, and P. Traverso, Specifying and analyzing early requirements in Tropos, Requirements Engineering, 2012, 9(2):132-150.

[3] Carlo Ghezzi, Dino Mandrioli and Angelo Morzenti, Trio: A logic language for executable specifications of real-time systems, Journal of Systems and Software, 2013, 12(2):107-123.

[4] L. Heitmeyer, R. D. Jeffords, and B. G. Labaw, Automated consistency checking of requirements specifications, Trans. Software Eng. Methodology, 2014, 5(3):231-261.

[5] Jackson, Alloy: a lightweight object modeling notation, ACM Trans. Software Eng. Methodology, 2014, 11(2):256-290.

[6] Z. Manna and A. Pnueli, The Temporal Logic of Reactive and Concurrent Systems, Specification, Springer, 2014.

[7] Madhuri et al., Reliable Security In Cloud Computing Environment, International Journal Of Information Technology, 2014, vol. 1, no. 1, pp. 21-28.

[8] Archana Pawar et al., A Review on Virtual Machine Scheduling in Cloud Computing, International Journal of Computer Science and Mobile Computing, 2014, vol. 3, no. 4, pp 928-933.

[9] Pranalini Joshi et al., Review on Fog computing: Reducing Insiders data theft attack in cloud computing, International Journal of Innovative Research in Advanced Engineering, 2014.

[10] Manoj Sharma et al., Evaluating Various Aspects of Cloud Computing Vendors with Comparison, International Journal of Research Aspects of Engineering and Management, ISSN: 2348-6627, 2014, Vol. 1, Issue 1, pp. 4-8.

[11] Kostas Stravoskoufos et al., A Survey on Approaches for Interoperability and Portability of Cloud Computing Services, International Journal of Research Aspects of Engineering and Management, ISSN: 2348-6627, 2014, Vol. 1, Issue 1.

[12] Chung Yung et al., Automatic Schedule Control for Distributed Software Development in Cloud Computing Environments, Journal of Industrial and Intelligent Information, 2014, Vol. 2, No. 1.

[13] Nick Papanikolaou et al., A Toolkit for Automating Compliance in Cloud Computing Services, Int. J., 2014, vol. 10.

[14] Muhammad Yousaf Saeed et al., Insight into Security Challenges for Cloud Databases and Data Protection Techniques for Building Trust in Cloud Computing, J. Basic. Appl. Sci. Res., 2014, vol. 4, no. 1, pp 54-59.

[15] Debabrata Sardda et al., A Mobile Cloud Computing Architecture with Easy Resource Sharing, International Journal of Current Engineering and Technology, 2014, vol. 4, no. 3.

[16] Rakesh Kumar et al., Open Source Solution for Cloud Computing Platform Using OpenStack, International Journal of Computer Science and Mobile Computing, 2014, vol. 3, Issue. 5, pp. 89 – 98.

[17] Sahaya Princy et al., A Study of Profit Maximization Based On Multiserver in Cloud Computing, International Journal of Advance Research in Computer Science and Management Studies, 2014, vol. 2.

[18] T.Swathi et al., Virtualization In Cloud Computing, International Journal of Computer Science and Mobile Computing, 2014, vol.3, Issue.5, pp. 540-546.

[19] Neethuanna Mathew et al., Application for Intra-College Communication Based on Cloud Computing, International Journal of Computer Science and Mobile Computing, 2014, vol.3, Issue.3, pp. 1012-1017.

[20] S Marston et al., Cloud computing - The business perspective, Journal of Decision Support System, 2011, vol. 51, pp. 176-189.
[21] D Sun et al., Surveying and Analyzing Security, Privacy and Trust Issues in Cloud Computing Environments, Journal of Procedia Engineering, 2011, vol. 15, pp. 2852-2856.
[22] Vengatesan K., and S. Selvarajan: Improved T-Cluster based scheme for combination gene scale expression data. International Conference on Radar, Communication and Computing (ICRCC), pp. 131-136. IEEE (2012).
[23] Kalaivanan M., and K. Vengatesan.: Recommendation system based on statistical analysis of ranking from user. International Conference on Information Communication and Embedded Systems (ICICES), pp.479-484, IEEE, (2013).
[24] K. Vengatesan, S. Selvarajan: The performance Analysis of Microarray Data using Occurrence Clustering. International Journal of Mathematical Science and Engineering, Vol.3 (2) ,pp 69-75 (2014).