Technical journals online.com

Research Paper

DYNAMIC CONGESTION CONTROL IN GROUP BASED SERVICE ROUTING FOR SCALABLE MANET

Pushpalatha S, Jaganathan P.

Address for Correspondence

Department of Computer Science and Engineering, PSNA College of Engineering and Technology, Dindigul, India.

ABSTRACT —

Service discovery is an essential application in Mobile Ad hoc Network. The scalability is a critical problem when designing service discovery protocols, since congestion occurs when many users frequently access the services. Therefore, service discovery protocols should be designed having the knowledge of congestion control in the network. In the proposed work, a non-Congested neighbour information is constructed by each mobile node to identify the congestion status of the nodes available in the Service Information cache. If packet loss occurs during the service invocation, alternate route is identified through non-Congested neighbour status from the Service Information Cache. The integration of congestion control in the service discovery protocol provides good performance under congested networks. The proposed work utilizes bandwidth efficiently and thereby saves energy.

INDEX TERMS—Service Discovery, Non-Congested Neighbour set, Congestion control, Mobile Ad hoc network.

I. INTRODUCTION

In Mobile ad hoc network (MANET), an autonomous group of mobile users communicate through bandwidth constrained wireless links. Service discovery plays a vital role in MANET to share the hardware and the software resources automatically on demand. Real time multimedia applications are frequently shared by many users, which increase the traffic in the network. This problem is critical when mobile nodes have limited transmission capacity and buffer capacity. Group based Service Discovery (GSD) protocol is one such a protocol which describes the services semantically using DARPA mark up language and to classify the service groups [1].

Forward Node Minimization enhanced Group based Service Discovery Protocol (FNMGSDP) is an extended version of GSD and it reduces the number of service request packets while discovering a service [2]. The cross layer service discovery protocols mentioned in [3] and [4] do not discuss about congestion. These protocols do not control the traffic created by the service invocation sessions. The congestion based approach reduces hit delay and increases hit rate for search protocol in unstructured peer to peer network. The downloading path from Service Provider to Service Client must be a reliable path. The popular services are frequently accessed by many users. Therefore, the loads among various service providers are to be evenly distributed [5]. The heavily loaded nodes cause congestion, which leads to packet loss and thereby the resources such as bandwidth and energy are wasted. Moreover, if the service is not identified using the local Service Information Cache (SIC), the service request is broadcasted in the network. This kind of broadcasting floods service request packets in the network. This may also cause congestion in the network. In addition to that, simultaneous service invocation sessions will create congestion in the dense and scalable network. The problems like long end-to-end delay, high overhead and low throughput occur due to congestion. This may be avoided by controlling the requests based on the congestion level of the nodes. Therefore, our aim is to build a cross layer service discovery protocol with the integration of routing and congestion control mechanism. The heavy network load is a problem closely associated with medium access and the packet forwarding mechanism [6-10]. The queue length based cross layer service discovery protocol is described in [11]. The congestion aware path leads to load balancing in the network [12]. Moreover, Quality of Service (QoS) based service selection protocols do not concentrate on congestion [13][14]. A new perspective of this problem might be to realize the congestion control in the network layer along with the service discovery to improve the performance. An Early Congestion Detection discovers non congested path between service client and service provider.

A. Early congestion detection and NHN Construction

The average queue length is used to estimate the congestion well in advance. The average queue length is calculated as

$$\text{avg\_que} = (1 - \text{w}_\text{q}) * \text{avg\_que} + \text{curr\_que} * \text{w}_\text{q} \quad (1)$$

In equation (1), w_q is the queue weight and is assigned 0.002 from RED queue [15].

Minimum and maximum threshold values for the queue length are used to control the traffic.

Minimum threshold = 35% of buffer size

Maximum threshold = 2 * Minimum threshold

$$\text{QueueStatus(QS)} = \text{curr\_que} - \text{avg\_que} \quad (2)$$

If QS > Maximum threshold, Congestion Status flag is set. The NHN setup is an initialization procedure and each mobile host periodically calculates its QS by using early congestion detection algorithm. A CSP (Congestion Status Packet with TTL =1) packet is broadcasted by the mobile nodes to its 1-hop neighbours to inform its congestion status. Then each mobile host constructs its NHN set consisting of 1-hop and 2-hop non congested nodes. So that the mobile nodes in the set can be used to forward broadcast traffic to minimize the flooding Traffic [10].

The objectives of congestion control integrated service discovery protocol are

(i) Improve successful completion of service invocation
(ii) Extend the lifetime of the nodes
(iii) Support QoS based service selection

(iv) Provide load balancing among the service providers

The proposed system is elaborated in section 2. The performance metrics and the simulation results are discussed in section 3. Finally, the conclusion is given section 4.

II. MATERIALS AND METHODS

The service discovery protocol is designed for MANET by integrating the service discovery with Adhoc On demand Distance Vector (AODV) routing protocol and to identify a non-congested path between Service Client (SC) and Service Provider (SP). The network congestion is reduced by minimizing the flooding of service request packets. Initially, 1-hop and 2-hop Non-Congested Neighbours (NCN) set is constructed to identify the congested nodes. The non congested neighbor information is updated in Congestion based Service Information Cache (CSIC). When the SC host wants to discover a service, it refers CSIC to find the non congested neighbor nodes and Service Providers to forward Congestion aware Service REQuest packet (CSREQ). The Congestion aware Service REPly (CSREP) packet sent by a SP node, confirms the downloading path as congestion free path. After the service discovery, Congestion aware Service Invocation (CSINV) packet is forwarded to the SP to download the service.

A. NCN Construction

In equation (1) the weight w_q is calculated as follows:

$$w_q = w_q \times \text{datarate} \times N_s \times N_{sv} \times N_{ss} \times \text{len(SRB)} \quad (3)$$

In equation(3), w_q is weight assigned from equation (1), data rate is transmission rate of a session, N_s is number of services available in the node, N_{sv} is number of services in the vicinity of the node, N_{ss} is number of current sessions and len(SRB) is number of CSREQ waiting in Service Request Buffer (SRB) of that node. The average queue length (avg_que) is calculated using equation (1) and queue status (QS) is calculated using equation (2). The factors affecting the network traffic are density of the service provider nodes, number of services in vicinity, current sessions count, data rate and number of CSREQ waiting in SRB of SP. Therefore, these parameters have to be considered for calculating w_q dynamically. As a result, the average queue length (avg_que) changes much slower than the current queue length (curr_que). The value of the Congestion Status Packet (CSP) is evaluated as follows:

If (QS < minimum threshold),
set congestion-flag =False
If (QS > minimum threshold and curr_que < maximum threshold),
set congestion-flag =True
If (curr_que > maximum threshold),
set congestion-flag =True
set CSP ={ node-id, congestion-flag }

Periodically, NCN construction is done to identify the congested nodes in a scalable network. Every mobile host broadcasts to its 1-hop neighbours about its congestion status using a CSP packet. So that, each mobile host learns its 1-hop non congested neighbour nodes and records the information in its non-congested 1-hop list. After that, each mobile host exchanges its 1-hop non-congested neighbour list with its neighbour nodes to learn its 2-hop non-congested nodes. Then, each mobile host constructs its NCN-set as a combination of 1-hop and 2-hop non-congested neighbour nodes. The NCN mobile hosts are used to forward CSREQ packets to minimize the flooding traffic. Each mobile host updates the non-congested node information in its CSIC table. It avoids unnecessary transmission of request packets in the congested network and waits for CSREP. The non-congested routes are predicted using CSIC and NCN.

B. Congestion aware Service Advertisement

Before forwarding a Congestion aware Service Advertisement (CSADV), the congestion status of the node is estimated by the nodes. If the congested nodes are identified from NCN set, the SP delays the CSADV packet transmission to reduce the traffic in the network. The nodes which have not received the CSDAV packet before the expiry time of the service, can extend the expiry time once by assuming that the CSADV packet is unable to be received due to congestion, but not due to mobility. In CSIC, the value of expiry time is multiplied by 2 and marked as congestion.

The fields in CSIC are

CSIC entry = {1 hop and 2 hop neighbour node, services, services in vicinity, congestion status, expiry_time}

C. Congestion aware Service Discovery

When the SC node wants to access a service, the SC checks its CSIC to find the availability of service. It determines the congestion free SP list from CSIC and forwards CSREQ to those nodes. In CSREQ packet, CTTL is set to 2. At each intermediate node, TTL and CTTL fields are decremented by 1 and the CSREQ packet is forwarded. When CTTL becomes 0, the nodes having the CSREQ packet again checks the CSIC to select non congested SPs and CTTL field is reset to 2 in CSREQ. This continues until the service is identified. When the intermediate nodes receive CSREQ, it checks whether the service is available in those nodes. If so, construct service reply (CSREP) packet and forward CSREP in the reverse route of CSREQ. The SP node responds to the first arrived CSREQ packet and sends back a CSREP through non-congested nodes. CSREP consists of non functional QoS parameters (remaining bandwidth and residual energy) and functional QoS. When the CSREP packet arrives at SC, less loaded SP is identified from congestion free path. If the service is unable to be identified from CSIC, CSREQ is broadcasted using congestion information alone. Finally the SC finds a non-congested path to SP. When the SC receives more than one CSREP, it selects the best one based on the QoS status.

QoS status= {Residual energy, Remaining bandwidth, functional QoS}

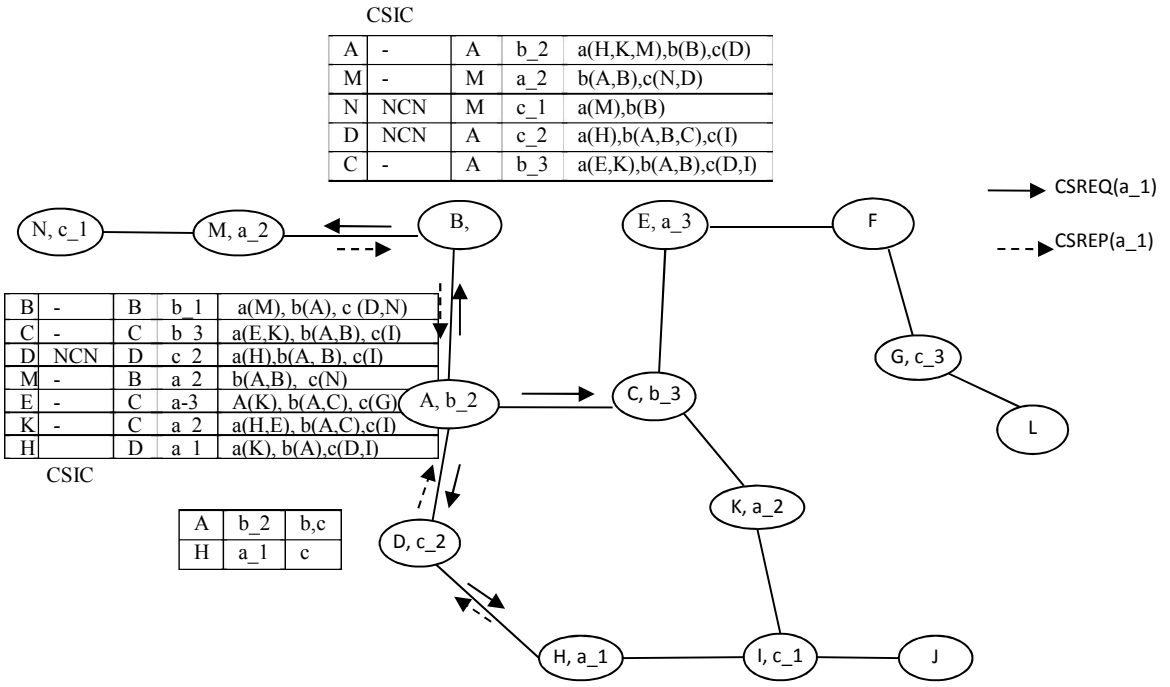

Figure 1 : DCC-FGSR Topology

In figure 1, the detailed CSIC is given for nodes A and B. 1-hop and 2-hop congested nodes list (N and D), which is collected by node B from NCN set and marked as NCN in CSIC. 1-hop and 2-hop congested nodes list of node A consists of D and H. When B requests a_1, it constructs the set of SPs providing a_1 service with NCN flag. The CSREQ packet is forwarded to those SPs (E,K,M) with CTTL =2.

The SPs provide service 'a_1' : { -}

The SPs provide service group 'a':{H,K,M,E}

2-hop Non Congested Node set : {E, K,M}

Therefore, CSREQ is forwarded to (E,K,M) with CTTL =2. At each intermediate node CTTL and TTL are decremented by 1. When CTTL expires and if the service is not identified, the 2- hop NCN set is checked again and the query is forwarded to the congestion free SPs. This continues until the service is identified or TTL expires. The service reply path is the downloading path from SP to SC, which is used for service invocation. Hence, the CSREP path must be a congestion free path for the service invocation. The service from the same group may be accepted in the congestion free path. In this example, only M sends CSREP to B.

This algorithm initially takes 2% extra delay for constructing NCN set, but the end-to-end delay is reduced (around 10–15%) for data transmission. So that the impact of NCN set construction delay is negligent. The NCN entries are valid for 30 seconds only. During this time, NCN need not be generated for any other CSREQ. In scalable network, the traffic must be controlled with respect to the network resources. Identifying reliable SP in a reliable path, satisfies the user requirements.

Algorithm NCN based Service Discovery
Begin

1. Case 1 : CSREQ Packet Forward process
2. Construct NCN set for all mobile hosts
   /* NCN – non congested nodes of the network */
3. For each CSREQ
4. {Hops=0; TTL=10; CTTL =2;
5. if (Service S_r is in 2-hop NCN list obtained from CSIC of node i)
6. { Forward CSREQ to non-congested SPs}
7. Else broadcast CSREQ to NCN nodes;
   //the required service is not available in CSIC }
8. case 2: Receiving CSREP Packet process
9. (while (TTL !=0 and S_r not found)
10. { if (the receiving node has service S_r)
11. {Construct CSREP packet
12. Forward CSREP to SC }
13. Else
14. {CTTL = CTTL -1; TTL =TTL -1
15. if (CTTL = = 0 and TTL! =0)
16. {set CTTL =2;
17. check NCN SPs
18. Forward CSREQ to SPs in NCN set }
19. Else if (TTL = =0)
20. {Drop CSREQ }}
21. End

D. Congestion aware Service Invocation

When more than one CSREP is received, the best SP is selected using QoS information from CSREP. The Service Request Buffer (SRB) of service provider can hold 10 CSREQ packets at a time. In order to reduce the waiting time of CSREP packets and to reduce the service invocation delay, the minimum value is chosen for SRB. If it exceeds, CSREQ will not be accepted. After the service discovery, Service Invocation (CSINV) packet is transmitted to the SP node. The SP node confirms the request in buffer and schedules the requests one by one. Hence, the proposed work reduces the overhead and automatically finds the non-congested path. Thus, this mechanism decreases the flooding of packets.

III. RESULTS AND DISCUSSION

A. Performance Analysis

1. Message complexity of service advertisements in 't' simulation seconds in 2 hop region.

$$A = \sum_{i=1}^{nSP} \text{nSADV} \times (1\text{hop} \,\&\, 2hop \text{ neighbours}) \quad (4)$$

Where nSP is number of service Providers.

2. Number of control packets used to forward service request in the network

$$B = \sum_{i=1}^{nSC} \text{nCSREQ} * (\text{NCN neighbours}) * \tau \quad (5)$$

'τ' – number of hops in the routing path

3. Number of control packets used for service reply in the network

$$C = \sum_{i=1}^{mSP} \text{mCSREP} * \tau \quad (6)$$

4. Total number of control packets in 't' simulation period is

$$A + B + C \quad (7)$$

5. Bandwidth usage

A×size (CSADV) +B × size (CSREQ) + C × size (CSREP) + size (data packets) (8)

6. Additional Control Packets during alternate route discovery:

$$D = \sum_{i=1}^{nSF} \text{nCSREQ} * (\text{NCN nneighbour}) * \tau \quad (9)$$

B. Simulation Results

The proposed work is simulated using NS2 [16]. The network consists of 100 nodes in a 1500, 1500 m terrine size. The radio range is 250 m with 2 Mbps bandwidth. The MAC layer uses IEEE 802.11 DCF (Distributed Coordination Function). The channel propagation model is random way point model. An interface queue at the MAC layer can hold 50 packets. At the network layer, routing buffer can store up to 64 data packets. The data flow uses CBR which varies from 4 packets to 16 packets and flows vary from 10 to 60 flows. The minimum speed and maximum speed of the node is 2m/s and 10 m/s respectively. The simulation time is 1000 seconds. There are 60% service providers and 3 service groups. The service advertisement interval is 20 seconds. The size of the service is downloaded from SPs is 100KB. The service discovery protocols GSR, CAA-GSR and DCC-FGSR are compared.

1. Hit rate for various service request frequency

The service requests are generated every 5 seconds and the number of accepted requests is shown in figure 2. The data rate is 4 packets/second. 30, 45 and 60 CSREQs are generated in three different scenario. 100% hit rate occurs up to 30 requests per second. 75% hit rate occurs for more than 30 requests and the hit rate gradually decreases when number of requests are increased gradually.

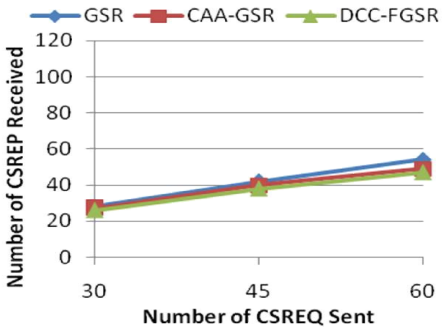

Figure 2: Service Hit Rate

2. Packet Delivery Ratio by varying the number of service invocation sessions

Packet Delivery Ratio (PDR) is the ratio between the number of packets received by the Service Client and the number of packets sent by the Service Provider. There are 3 scenarios for generating 30, 45 and 60 service requests in 1000 seconds. The data rate is 4 packets/second. The packet delivery ratio is shown in figure 3. When the offered load is below 16 requests, there is no packet loss. When the number of flows increases from 30 to 50, more CSREQ packets are generated and transmitted. This leads to high consumption of the node's buffer and causes packet loss. When the number of CSREQ is restricted per unit time, the number of packet drop can also be restricted. DCC-FGSR resolves congestion by finding alternate path using NCN set. Therefore, DCC-FGSR has better packet delivery ratio than CAA-GSR.

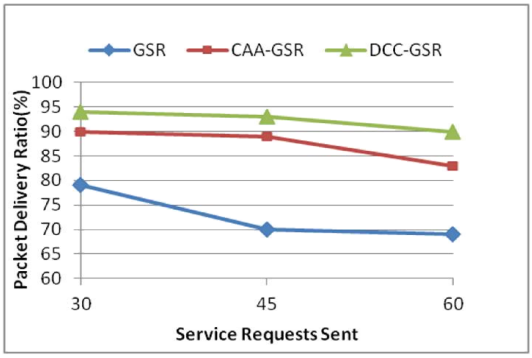

Figure 3: Packet Delivery Ratio

3. Routing Overhead:

There are 3 scenarios for generating 30, 45 and 60 service requests in 1000 seconds. Figure 4 shows the control packets generated by various service providers. When the offered load is below 25 flows, DCC-FGSR do not show better performance than CAA-GSR. When the offered load is low, the network congestion level falls in safe zone.

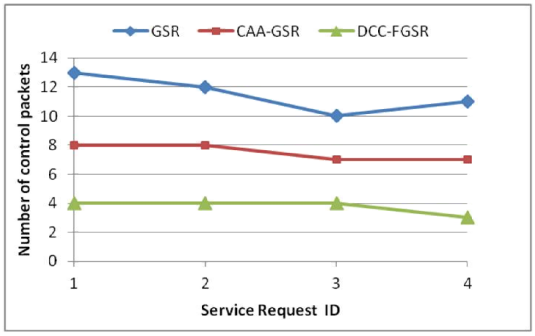

Figure 4: Control Overhead

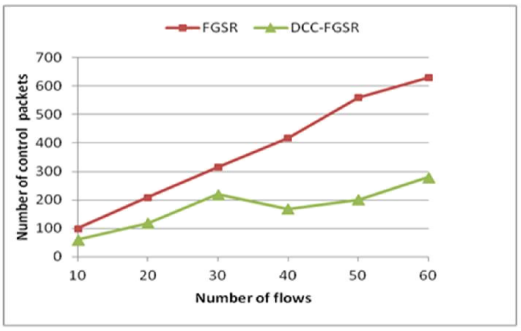

Figure 5: Control Overhead for various Flows

When the service request is increased from 30 to 50 flows, DCC-FGSR incurred heavy routing overhead and consumed more control packets to find an alternate path. The DCC-FGSR consumed control packets 8% less than CAA-GSR. DCC-FGSR is unaffected by increasing the traffic, because it resolves congestion by using NCN set to find alternate routes. This is described in figure 5.

4. The End-to-End Delay of Service Discovery

When the number of flows occurs below 16, the congestion level falls in safe zone. Figure 6 represents the end-to-end delay comparison for cross layer FNMGSDP (FGSR) and DCC-FGSR**.** When 20 requests are allowed and the network congestion level falls to congestion zone, the end to end delay of DCC-FGSR increases almost linearly with an increased offered load. For 30 requests, DCC-FGSR demonstrates 6% reduction of the delay over the FGSR.

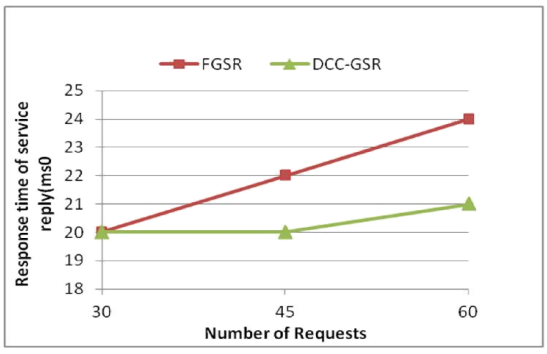

Figure 6: End-to-End Delay of Service Discovery

5. The End-to-End Download Delay:

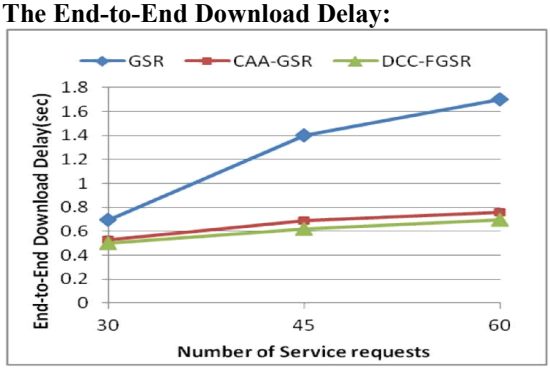

Figure 7: End-to-End Download Delay of Service Discovery

The downloading delay from SP to SC is shown in figure 7. The average end to end download delay is analyzed for GSR, CAA-GSR and DCC-FGSR by varying service size and data rate. When the number of CSREQ is limited to each SP, the end-to-end download delay is less than FGSR. When the number of flows increases from 30 to 50 user sessions, CSREQ packets are generated and transmitted. During the service invocation session, whenever the packet drop is noticed, alternate path is discovered. A non-congested alternate path is identified from 2-hop lists and establishes a route to destination immediately. This is due to the fact that NCN information is available in CSIC, few control packets are required to establish a new path. For high offered loads (between 40 and 50 flows), the delay is reduced by 20% over FGSR.

6. Energy Consumption

The simulation is done for 1000 seconds with 100 nodes. The residual energy of some random nodes is calculated for 5 times and the average is calculated. The service requests are generated periodically and the average of energy consumption is calculated. This is shown in figure 8. The initial energy of each node is 100 joules. The service provider nodes are continuously servicing the clients. From figure 8, it is shown that the energy is almost evenly consumed by all SPs and the lifetime of the nodes is approximately the same except very few nodes. It is assumed that the applications would provide better performance when the service discovery is integrated with the cross layer information. Since the CSREQ and the CSREP packets are forwarded in the congestion free network, the selection of the same SP is avoided and the SPs might be alternatively chosen. This balances the energy of the nodes and increases the lifetime.

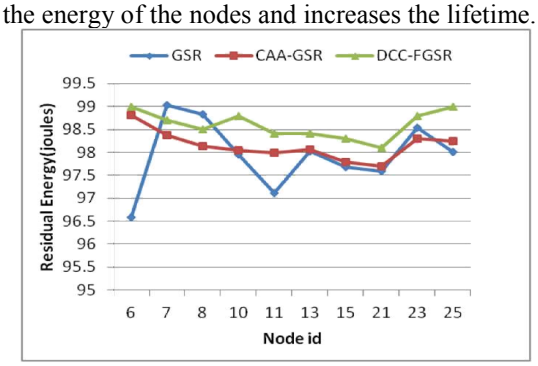

Figure 8: Residual Energy of nodes

IV. CONCLUSION

The cross layer Service Discovery Protocol is designed with the knowledge of congestion status of the mobile nodes. The Non-Congested Neighbour set of two hop region is constructed by each mobile node to identify the congestion status of the nodes available in CSIC. Before forwarding any service discovery packet, NCN status is verified to avoid overloading in the network. If packet loss occurs during the service invocation, alternate route is identified using NCN set. DCC-FGSR shows the improved packet delivery rate, control overhead and the end-to-end download delay than CAA-GSR. The remaining energy level of the nodes shows the load balancing among the nodes.

REFERENCES

1. Dipanjan Chakraborty, Anupam Joshi and Yelena Yesha, “Toward Distributed Service Discovery in Pervasive Computing Environments”, IEEE Transactions on Mobile Computing, 2006, Vol.5,No.2, “doi.ieeecomputersociety.org/10.1109/TMC.2006.26”.
2. Zhenguo Gao, Ling Wang, Mei Yang and Jianping Wang, “FNMGSDP: An Optimized Group-Based Service Discovery Protocol for MANETs”, Springer Wireless Personal Communication, 2009, 57:137-162, “doi:10.1007/s11277-009-9849-2”.
3. Dipanjan Chakraborty, Anupam Joshi and Yelena Yesha, “Integrating service discovery with routing and session management for ad-hoc networks”, Elsevier Ad Hoc Networks, 2006, Pg.No: 204–224. “doi:10.1016/j.adhoc.2004.03.016”.
4. Zhenguo Gao, Yongtian Yang, Ling Wang, Jianwen Cui and Xiang Li, “FNSCSDP: A forward node selection based cross-layer Service Discovery Protocol

for MANETs", Springer verilog, 2006, LNCS 4325, pp 220, "doi:10.1007/11943952_19".

5. KinWah Kwong, Danny H.K. Tsang, "A congestion-aware search protocol for heterogeneous peer-to-peer networks", Journal of Supercomputing , 2006, 36:265–282, "doi: 10.1007/s11227-006-8297-y".
6. Mikhail Badov, Anand Seetharam, Jim Kurose, "Congestion-Aware Caching and Search Information-Centric Networks" ACM International conference on Information Centric Networking, 2014, "doi: 10.1145/2660129.2660145"
7. Kaouther Abrougui, Azzedine Boukerche and Hussam Ramadan, "Efficient load balancing and QoS-based location aware service discovery protocol for vehicular ad hoc networks", Eurozip journal on Wireless Communication and Networking, 2012,, p. 96, "doi:10. 1186/1687-1499-2012-96".
8. S.Santhosh baboo and B.Narasimhan, "A Hop-by-Hop Congestion-Aware Routing Protocol for Heterogeneous Mobile Ad-hoc Networks," (IJCSIS) International Journal of Computer Science and Information Security, 2009.Vol. 3, No.1.
9. SenthilKumaran and T.Sankaranarayanan V. "Early detection congestion and self cure routing in MANET", In: Proceedings of Springer LNCS computer and information science, 2011, vol. 142. p. 562–7, "doi: 10.1007/978-3-642-19542-6_110".
10. T.SenthilKumaran and V.Sankaranarayanan, "Early congestion detection and adaptive routing in MANET", Egyptian Informatics Journal, Volume 12, Issue 3, November, 2011, Pages 165–175, " doi:10.1016/j.eij.2011.09.001".
11. S.Pushpalatha and P.Jaganathan, "Congestion aware and Adaptive Service Discovery for MANET", Australian Journal of Basic and Applied Sciences, 9(20), pp 265-272, 2015.
12. Puri and S. R. Devene, , "Congestion avoidance and load balancing in AODV-multipath using queue length", in Proceedings of the 2nd International Conference on Emerging Trends in Engineering and Technology (ICETET '09), pp. 1138–1142, Nagpur, India, December, 2009, "doi: 10.1109/ICETET.2009.62".
13. Hassan Artail, Haïdar Safa, Paul Salameh, Salim Chedrawi and Pierre El Khoury, "Quality-of-service-aware cluster-based service discovery approach for mobile ad hoc networks", Int. J. Communication Systems 27(11), 2014, 3107-3127, "doi: 10.1002/dac.2529".
14. 14 E. Christopher Siddarth and K. Seetharaman, "A Cluster Based QoS-Aware Service Discovery Architecture Using Swarm Intelligence", Communications and Network, 2013,Vol. 5 No. 2, pp. 161-168, "doi: org/10.4236/cn.2013.52018
15. Floyd S and Jacobson V, "Random early detection gateways for congestion avoidance". IEEE/ACM Trans Networking 1993;1(4):397–413.
16. The Network Simulator, ns2 , http://www.isi.edu/nsnam/ns.