International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 17 (2015) pp 37839-37845

Design and Implement of Light Weight QoS Algorithm for Ad hoc Networks

Seyed Hossein Hosseini Nazhad Ghazani
Department of Computer Engineering, Ahar Branch, Islamic Azad University, Ahar, Iran
S.HosseiniNazhad@iau-ahar.ac.ir

Rasim Alguliyev
Department of Computer Engineering, Azerbaijan National Academy of Science Baku, Azerbaijan
Rasim@science.az

Abstract
Ad hoc networks attract a considerable attention due to their simplicity, low cost, and efficiency. Obviously, using multimedia applications on these networks will be more widespread in the future. In contrast with best-effort flows, real-time ones seriously need Quality of Service (QoS) support. This article proposes a completely distributed algorithm to provide QoS in ad hoc networks. The proposed algorithm, entitled Light Weight distributed QoS Algorithm (LWQA), using queue data structure and linear algebra, dynamically adjusts Contention Window (CW) related to the flows. While serving the flows, it selects the flow with the smallest CW in a node and then serves it. Furthermore, LWQA utilizes static or low speed nodes during routing of real-time flows to increase their QoS. In addition to considering flows' priorities, the proposed algorithm is able to distinguish between flows of the same type. It is noteworthy that all these items might be done without imposing heavy overloads and without using resource reservation methods. We implemented the proposed algorithm in a simulation environment using the Network Simulator Version 2 (NS-2) software. The simulation results demonstrate that LWQA improves the QoS in ad hoc networks.

Keywords: Wide Area Ad hoc Networks, Quality of Service, Back-off Time, Contention Window, Delay, Real-time Flows.

Introduction

In modern world networking, the computational resources and data sharing are necessary to achieve fast accessibility. Due to the rapid growth and connection capability of electronic equipment, the utilization of ad hoc network is increasing. As ad hoc networks are spreading considerably, they are exploited for data sharing and transmission of multimedia applications as well as normal uses [1, 2]. The applications require different level of Quality of Service (QoS) provision to them in order to meet user satisfaction. However, the mobility of the communicating nodes, leading to rapidly changing network topology, make QoS support a very complex process. Several distributed algorithms, while imposing minimum overload on the network and attempting to perform the tasks with the minimum cost and efficient resource utilization, have been designed for QoS support in ad hoc networks. Some algorithms exploit static parameters such as CW size, frame size, and inter-frame space (IFS) for different type of flows to provide QoS in ad hoc networks [3,4]. The weaknesses of these algorithms are due to lack of attention to the current status of the network, flows, and lack of optimal use of available resources in the network.

As opponents of using static parameters, some algorithms dynamically assigned the priority of flows to support QoS in ad hoc networks, but neglecting the current status of flows [5-8]. Therefore, these algorithms are unable to distinguish between QoS requirements of the same type flows. On the other hand; some QoS supporting algorithms are usable in the networks, where the number of real-time flows does not exceed a specified range [9-11]. The weakness of them lies in their poor admission control; therefore, QoS of existing flows is degraded by contention to new flow. Along with the admission control utilization, resource reservation is one of the QoS support solutions in ad hoc networks [12-15]. But, the resource reservation method has its challenges i.e. some control signals are utilized for resource reservation. These signals compete with data packets inside the network; and as a consequence of collision between control and data packets, the overall system efficiency decreases. Node mobility is another problem of this method. It is essential that the source node reserve the resources to send high priority data packets. As the nodes are mobile, after reservation, they may move and exit resource reservation area. Therefore, it causes the reserved sources to become useless for a time span which, in turn, decreases network efficiency. Furthermore; some existing algorithms, during routing process, do not consider the speed of nodes [16-20]. Therefore, high speed nodes have the equivalent probability of incorporation in a communication path as static or low speed nodes. It is evident that the paths, composed of high speed nodes, have lower stability and are probable to be broken. Among them, MFQMAC [20] assures QoS through service differentiation among different classes of flows and provides fairness among traffic flows of same priority class. The major problems of MFQMAC are as follows: it does not consider the speed of nodes during routing process, the current status of the flows in calculation of back-off time, and the current status of the network for admission control.

In the following part, the proposed algorithm is introduced. It overcomes the mentioned problems in a distributive manner using dynamic calculation of both CW and back-off times with regard to the flows' statuses as well as the current status of the network, and it utilizes static and low speed nodes during routing real-time flows.

Materials and methods

Transmission of real-time flows over the ad hoc networks is usually interrupted by packet loss, caused by interference [21] and also, much more resources are required to control these flows; consequently, in order to make optimal use of network resources, transmission of real-time flows must be done with greater accuracy, minimal loss, and collision. The accuracy of QoS support in Light Weight distributed QoS Algorithm (LWQA) is increased by:

(1) QoS support by creating valuable paths,
(2) QoS support regarding to network status, and
(3) QoS support using dynamic adjustment of CW.

A. QoSsupport by creating valuable paths

Link break is one of the most important causes of QoS decrease in ad hoc networks, which leads to partial loss of packets. Once links are broken, both routing and transmission processes of lost packets are repeated, leading to an increase in delivery latency of the packets and a decrease in the QoS of the flows. In ad hoc networks with longer communication paths, it is more frequent to witness the link break. Due to lower possibility of both route break and QoS decrease in links with static or low speed nodes, they are more valuable and efficient compared to those having high speed nodes. In this article, a path is regarded as valuable, when it consists of static or low speed nodes. It can be argued that, it is better to route all flows with valuable paths, since they provide the most stable paths. Nevertheless, considering the number of existing flows in the ad hoc networks, it is less possible to use mentioned paths in order to transmit all flows, which leads to decreased efficiency of utilizing network capacity. Therefore, as shown in Figure 1, in order to transmit real-time and best-effort flows, valuable and ordinary paths are preferred, respectively. In the ordinary routing, a link consists of static, low, and high speed nodes.

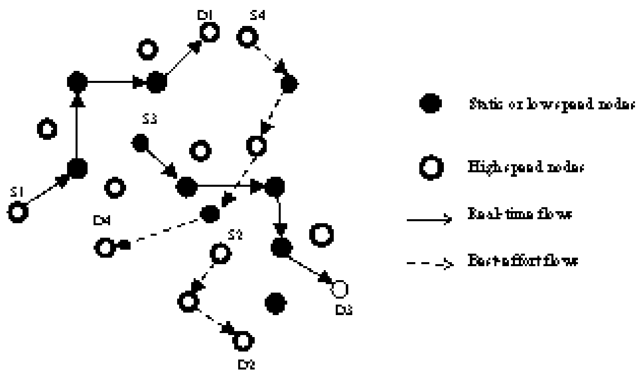

Fig. 1. Valuable and ordinary paths for transfer real-time and best-effort flows

To implement the explained idea; considered routing algorithm should be modified in a way that, it can be capable of making distinctions among nodes by considering their movement rates. In this article, Ad hoc On-demand Distance Vector (AODV) has been used as a routing algorithm [22]. In order to route discovery for real-time flows, with a small modification in the mentioned algorithm, only static and low speed nodes respond the request by either forwarding the Route Request (RREQ), or unicasting the Route Reply (RREP) packets. In the modified AODV routing algorithm, the high speed nodes do not respond to the RREQ, unless they are the destination nodes (see Figure 1). In transmitting the real-time flows; if a valuable path is not discovered in the first effort, transmitting node can either wait for some time and retry, or decrease the QoS level of required service. In this article, we used the first method due to its simplicity. We also utilized the ordinary AODV to route the best-effort flows. It is worth notifying that, in this article, the routing management is done by a module entitled Create_Safe_Path. Therefore, this module is part of routing process and it calls for route discovery, in a way that regarding types of data packets, the module uses either the modified AODV for real-time flows, or the ordinary version of it for best-effort flows. Therefore, the main goal of this module is to discover a valuable transmission path having satisfied QoS requirement, along with the cooperation of admission control module. The specification of admission control is considered following the explanation of modified AODV algorithm.

The modified AODV routing algorithm consists of two sections; Request and Reply. In the request stage, the source node transmits the RREQ messages for the new flow, as shown in Figure 2. These messages, indicated by solid arrows, include QoS information such as type of flow, needed quality for flow, and the minimum operational power or accumulated delay in all previous nodes. Each intermediate node writes information of RREQ packet in its routing table as soon as receives it. As long as the average node speed for providing services is appropriate for received type of the packet, and basicallyif the flow is locally acceptable in receiver node, it will rebroadcast the RREQ message. In this article, the average rate of nodes' speed, in order to reply to the real-time flows, is considered to be 0.5 m/s. Meanwhile, there have not been any limitations considered in transferring the best-effort flows. On the contrary, if the intermediate node is not able to fulfill flow requirements, it eliminates RREQ packet. The intermediate nodes announce the potential load by broadcasting Neighbor Reply (NREP) messages, which are illustrated by dotted lines in Figure 2. The RREQ packet will reach the destination node if there exists a path with the required quality.

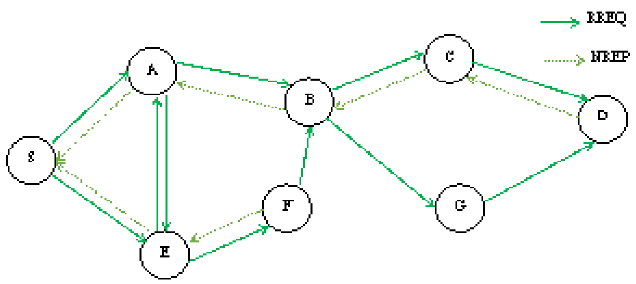

Fig. 2. Request Stage

In the reply stage, the destination node transmits RREP message in an inverse direction toward the source node, as shown in Figure 3. In this stage, intermediate nodes update the load information of the neighbors through NREP messages transmitted in the request stage. As a result, they will be able to recalculation the predicted QoS of flows more accurately and forward the RREP message if the new flow is locally acceptable. The source node will choose an optimal path according to the quality of candidate paths to transfer the flows. The nodes, existing in the selected path, transmit NREP packets to their neighbors in order to verify the status of accepted flows. In this way, all nodes affected by the new flow will receive updated information about channel consumption.

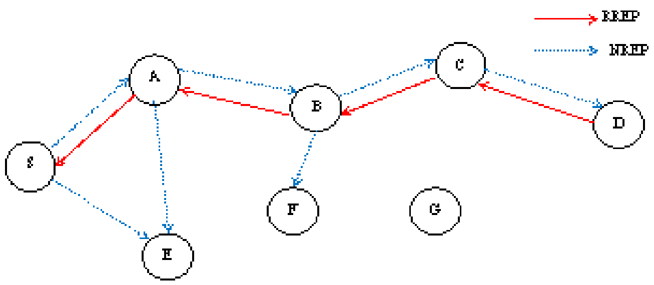

Fig. 3. Reply Stage

As described, to transfer a real-time flow, a valuable transmission path with satisfied QoS requirement is needed. In order to create it, Create_Safe_Path uses the modified AODV routing algorithm and cooperates with admission control. The latter is used to avoid admitting flows beyond the network capacity. The admission control needs to estimate channel consumption exactly and predict quality of the flows, such as operational power and transmission latency, to perform its' tasks. In ad hoc networks, all nodes around a particular node compete on accessing the channel. Therefore, the shared nature of communication channel makes it difficult to estimate channel consumption and predict quality of the flows. In this article, the estimation is done by using a Model-Based Resource Prediction (MBRP) mechanism [23]. This mechanism calculates the operational power and latency of flows by sampling behavior of nodes' back-off. The MBRP tries to predict the quality of the flows for both progressing and new traffics, so that an appropriate decision could be made regarding acceptance or rejection of the new flow based on quality control and management policy. The decomposition input in MBRP is a set of flows such as A = {a1, a2, .., as} in the network, where s is the number of priority classes supported by the system and ai denotes the number of flows inside ith class. The output of this mechanism is considered to be the calculation of an average delay or operational power for each flow. To address issues such as hidden terminals and unexpected collisions during the run time, MBRP utilizes the results of flow progress measurements as a feedback. It should be noted that the estimation function is analyzed and executed locally. This function located in MAC layer could control the behavior of packet scheduling more properly and provide precise prediction regarding reservation information at the Internet Protocol (IP) layer.

B. QoSsupport regarding to network status

An algorithm managing and controlling access of nodes to common channel, plays a prominent role in regulating smart transmission of flows. In this article, the scheduling algorithm, utilized for managing and controlling the nodes' access to the channel is based on IEEE 802.11 [24]. There is a variety of algorithms providing various services based on IEEE 802.11. They consider diverse values of back-off time, IFS, and CW for different types of traffics. For example, higher back-off is utilized for low priority traffics, while a lower one is utilized for high priorities. By employing aforementioned methods, better services could be provided for high priority flows in comparison with low priority ones. However, by investigating these methods, one may conclude that all these parameters are considered statically. Static parameters lead to a lack of adaptation to network traffic, resulting in efficiency decreased. It is difficult to find appropriate static values to adjust different types of traffics' parameters, since these parameters must be changed based on network status. The LWQA has overcome this problem by considering dynamic parameters and noticing the network status. If the network status is included in back-off time calculation, flow transmission would be smarter, and the QoS would be improved. Therefore, if each node detects the network status as saturated, it will set a higher back-off time for itself and avoid competing with the flows which are being transmitted. These nodes allow current flows to utilize network resources by assuming higher back-off time. On the other hand, a node will choose a lower back-off time if it detects the network status as uncongested. Therefore, the node utilizes network resources; consequently, it prevents resources' waste. To include the network status in the back-off time calculations, Rcol, r, and pri parameters are added to IEEE 802.11:

$$Back-off = Rand[0,(2^{r} + R_{col} \times pri) \times CW] \times Slot\_Time \qquad (1)$$

Where R_{col} parameter denotes the number of collisions occurred between two successful transmissions in one node, pri and r are constant values chosen according to the priority level of data packets. As it is clear, when the network is saturated, both the number of collisions and R_{col}value increase. Any increase in R_{col}, increases the back-off time of the flows. Consequently, the number of contentions, in saturated network, decreases and the QoS is maintained at an acceptable level. Conversely, when the number of existing flows is small, the values of both $R_{col}$and back-off time decrease; as a result, it maintains the efficiency of the system in a high level. Considering the Eq. (1), back-off value increases or decreases linearly, i.e. all flows of the same type are treated in the same way. To solve this problem, the current status of each flow must be considered while calculating the back-off time values, as it has been explained below

C. QoSsupport using dynamic adjustment of CW

Since appropriation of static values to CW leads to a relative differentiation between flows, a better differentiation can be created between them by dynamic regulation of CW. In this article, we have used current status of flows in order to calculate CW in a dynamic state. To manage the current status of flows, we create queues in the nodes. It is performed in each node, completely locally and without imposing any overload. As shown in Figure 4, for each real-time flow, one queue is generated, and for all best-effort flows just one queue is formed by passing one node.

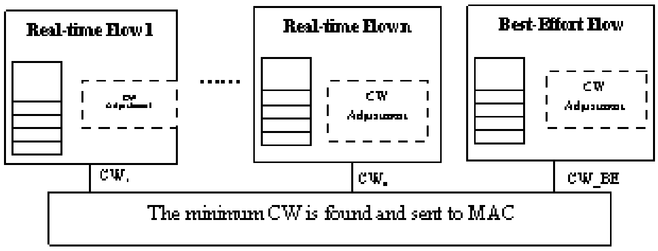

Fig. 4. Queues in the nodes

Each node investigates the packet as soon as receives it. If the received packet is a real-time type, the node puts it in the associated queue. Otherwise, it is inserted in the queue, which is appropriated to best-effort flows. Along with the data packets, type of packets, arrival time of them to queues, and the number of existing nodes along the path would be saved in queues. Regarding saved information, and since the requirements of existing flows are different with respect to the type of flows, various methods should be employed to adjust the CW. For this purpose, the existing flows are divided into three groups: delay-sensitive flows, that are mostly conversational, need their data packets to reach the destination within a specific time; bandwidth-sensitive flows, that need operational power; and best-effort flows, that are resistant to changes in bandwidth and delay, and do not require a specific QoS. The methods of CW adjustment are provided for the groups mentioned as follows.

i. CW adjustment for delay-sensitive flows

The main QoS parameter regarding delay-sensitive flows is end-to-end (d) packet delay. These flows should have passed through their path within a determined time span; otherwise, it is assumed that they did not reach their desired QoS. To meet the requirements of these flows, the number of nodes (m) between source and destination should be known. By achieving this value, and dividing d by m, the allowable delay value for each data packet, to wait in each node, is derived. Due to the utilization of the AODV routing algorithm, the number of nodes is known. Hence; each node tries to limit the delay time of packets below or equal to d/m, by locally adjusting the CW. As a result, the sum of delays in all nodes along the path, becomes smaller than d. It is noteworthy that the allowable delay value is carried by a number of initial packets of each flow; therefore, all nodes along the path will be informed about this value. The Eq. (2) is utilized to adjust CW of delay-sensitive flows. Considering this equation, if the delay of a node exceeds the allowable delay value, the CW decreases. Subsequently, the back-off time decreases as well. Afterwards, the desired node takes the control of the channel, and sends the flow. On the other hand, if the delay is less than the allowable delay value, the CW increases. As a result, the back-off time increases and the flow stops competing. Hence, the network resources would be free and other flows would utilize the resources.

$$CW^{(n+1)} = CW^{(n)} * (1 + a \frac{d/m - D^{(n)}}{d/m}) \tag{2}$$

Where n denotes the n^{th} update; D is a real packet delay in the node, and a is a small positive value, which experimentally is considered to be 0.1 in this article. This equation moderates the CW value of delay-sensitive flows so that the delay of a packet in each node would be less than the allowable delay value.

ii. CW adjustment for bandwidth-sensitive flows

Operational power is the main QoS parameter of the bandwidth-sensitive flows. According to queuing theory, the operational power of a flow might be maintained constant by the length management of the queue. In other words, constant length of the queue demonstrates that the input and output rates of the packets are equal and, the required operational power is met. The Eq. (3) is utilized to adjust CW of the bandwidth-sensitive flows. The major goal of this equation, is to maintain a constant length for the queue and provide operational power.

$$CW^{(n+1)} = CW^{(n)} + \beta \times (q - Q^{(n)}) \tag{3}$$

Where n denotes the n^{th} update; q is the threshold value of queue length, Q is the real length of the queue, and β is a positive value. If the real length of the queue is larger than the threshold, the desired QoS is not fulfilled. Therefore, this equation reduces CW which, in turn, increases packet transmission. It compensates for previous shortcomings. In contrast, when the length of the queue is less than the threshold, it increases CW. As a result, the back-off time would increase and packets' transmission rate would decrease. When the length of the queue varies in the vicinity of the threshold value (q), the average of flow's operational power is roughly equal to required operational power. Regarding the guidelines given in [25], q was set to be equal to 5 in this article.

iii. CW adjustment for best-effort flows

Best-effort flows are not sensitive to the level of provided services. Therefore, all packets of these flows are kept in a common queue. The CW values of these flows are regulated to avoid congestion of these packets in the network. Moreover, it guarantees that non-best-effort flows are properly served. While sending the best-effort flows, the network status is investigated. If more important flows are passing through the network, the best-effort flows stop activity. Nevertheless, the waiting time should not be so much that these flows are accumulated. Thus, the CW of the best-effort flows is adjusted using the following equation:

$$CW^{(n+1)} = CW^{(n)} \times (1 + \gamma \times (f - F^{(n)})) \tag{4}$$

In Eq. (4), n denotes the n^{th} update; f denotes the congestion threshold value when the channel is free, F is the real free time of the channel, and γ is a positive value. The real free time of the channel is the average time between two time intervals through which the channel is busy. In this equation, F would be smaller than f, when the real-time flows are transmitted.

Evaluation of LWQA

To evaluate the performance of LWQA in supporting QoS, we compare it with MFQMAC [20] using the Network Simulator Version 2 (NS-2) [26]. In our simulations, the routing algorithm, channel bandwidth, and nodes movement rate were AODV, 11 Mbps and 0 to 3 m/s, respectively. The evaluations demonstrate acceptable performance of LWQA in improving QoS for network users. Following table presents the parameters used in our simulations.

TABLE.1. Simulation Parameters

Parameters	Value
a	0.1
β	1
f	1 ms
r	0.1
q	5 Packets
pri for real-time flows	0.5
pri for best-effort flows	1
Update interval of CW	0.1 s

A. Evaluation of LWQA and MFQMAC

In this section, we evaluate LWQA and MFQMAC's ability to keep QoS guarantees to delay-sensitive flows. The simulation area is considered to be 1000 m * 1000 m. In this simulation, 8 delay-sensitive, 8 bandwidth-sensitive, and 8 best-effort flows are utilized, which start in the first 115 seconds of the simulation. These flows generate 50 512-byte packets/s. Each delay-sensitive flow should be served in order to reach the destination in 100 ms. The source and destination of the flows are randomly chosen from 150 nodes inside the network. The number of nodes that each flow needs to pass is considered as a random value between 1 and 7 diametric. Figure 5 shows the average delay of delay-sensitive flows. The delay requirement of these flows is indicated by the dotted line in this figure. Both algorithms have kept the delay of flows less than the required value; nevertheless, LWQA demonstrates better performance because of two reasons. First, it utilizes static and low speed nodes during routing non-best-effort flows. Second, it considers the flows' statuses as well as current status of the network in calculating back-off time. It can be assumed that LWQA is able to manage networks with larger number of nodes. To verify this assumption, these algorithms were evaluated once more.

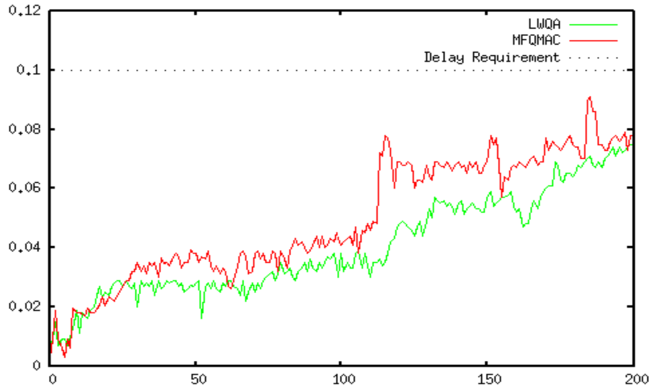

Fig. 5. Average Delay of Delay-sensitive Flows

B. Evaluation of LWQA and MFQMAC in Large Scale Network with Increase in Number of Nodes and Flows

The area in last experiment is 2000 m * 2000 m and 180 nodes are randomly distributed. The numbers of delay-sensitive, bandwidth-sensitive, and best-effort flows are 20, 25, and 25, respectively. The source and destination nodes, chosen randomly from existing nodes, communicate with each other. As the goal of this experiment is to investigate the behavior of algorithms in the networks with large number of nodes, the packet size and rate of the flows are considered to be like the previous experiment. In the first 50 seconds of this simulation, 4 real-time, 4 best-effort, and 4 bandwidth-sensitive flows were transmitted between nodes of the network. Figure 6 and Figure 7 show the average delay of delay-sensitive flows, and the violation of bandwidth guarantees to bandwidth-sensitive ones, respectively. As shown, LWQA controls the delay of the admitted delay-sensitive flows below their required time (100 ms) and shows no violations to the bandwidth guarantees. On the other hand, regarding MFQMAC, when other residual flows are transmitted within 50 and 80 seconds of the stimulation and network load increases, admits too many flows in a way that both delay and bandwidth are degraded. Nevertheless, LWQA keeps QoS guarantees for these flows. It achieves this goal by considering admission control, network status, and current status of flows in back-of time calculation; also, it is the case by using static and low speed nodes during routing process.

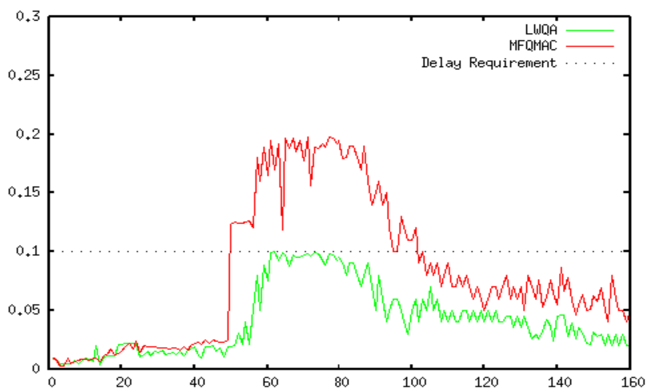

Fig. 6. Average Delay of Delay-sensitive Flows

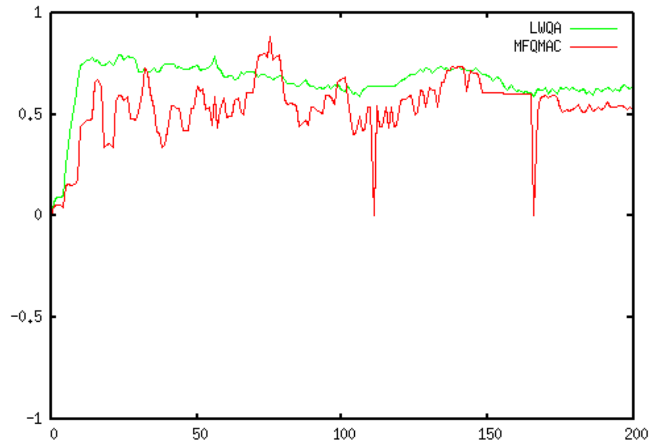

Fig. 7. Bandwidth Guarantees to Bandwidth-sensitive Flows

In this simulation at time 80 s of the simulation, 4 delay-sensitive, 4 bandwidth-sensitive, and 4 best-effort flows come to end, leading to a decrease in the number of existing flows. As a result, MFQMAC gradually obtains its control over the flows and provides them with the required QoS. But the time needed for MFQMAC to return to the normal condition is more than that the time needed by LWQA. Therefore, LWQA provides more stable packet delay and operational power than MFQMAC.

Conclusion

In this article, we introduced a QoS support algorithm entitled LWQA. The proposed algorithm aims to improve QoS in ad hoc networks. Several algorithms have been introduced, but the proposed one is more advantageous regarding different perspectives. Firstly, it does not need resource reservation to provide QoS. Secondly, it classifies the flows and provides different services in accordance with their requests. This is the common characteristic of various algorithms; nevertheless, the novelty of LWQA lies in the ability to provide different types of services for flows of the same type. Thirdly; in most QoS supporting models, all flows compete with each other to obtain their needed QoS. But in LWQA, a flow avoids contending others and tries to improve their QoS if it achieves required QoS. Finally, this algorithm distinguishes between fast and slow nodes. It utilizes valuable path, consisting of slow and static nodes, to send important data packets. Through simulations, we compared the performance of LWQA with MFQMAC and revealed the ability of LWQA to improve the QoS for ad hoc networks.

References

[1] Jain, M., Kriti, P. and Rakhee, "Optimal Channel Allocation in Cellular Systems," Journal of Information and Communication Technology, pp. 41-64,2002.

[2] Madhu, J., Vidushi, S., and Kriti, P., "Adaptive Batching Scheme for Multicast Near Vido-on-Demand (NVOD) System," Journal of Information and Communication Technology, pp. 1-12,2006.

[3] Hannan, X., Seh, W. K., Anthony, L., and Kee, C. C., "A Flexible Quality of Service Model for Mobile Ad-hoc Networks," Proc. IEEE 2000 Vehicular Technology Conference, Tokyo, pp. 445-449,2000.

[4] Mangold, S., Choi, S., and May, P., "IEEE 802.11e Wireless LAN for Quality of Service", Proc. of the European Wireless, Florence,Italy, pp. 32-39,2002.

[5] Wang, P., and Jiang, H., "A New MAC Scheme Supporting Voice/Data Traffic in Wireless Ad hoc Networks," IEEE Transactopn on Mobile Computing pp. 1491-1503,2008.

[6] Rajeshkumar, R., "Bandwidth Allocation and Routing Information for Wireless Mobile Ad-hoc Networks,". Proc. IEEE 2010 Conference on Fuzzy Systems and Knowledge Discovery, China, pp. 2968-2972,2010.

[7] Budyal, V., and Manvi, S., "Intelligent Agent Based Delay Aware QoS Unicast Routing in Mobile Networks," Journal of Multimedia and Ubiquitous Engineering, pp. 11-28,2013.

[8] Elizarraras, O., Panduro, M., and Méndez, A., "Reyna, A. MAC Protocol for Ad Hoc Networks Using a Genetic Algorithm," The Scientific World Journal, pp.1-9,2014.

[9] Qiao, Y., Bing, S., Xiao-bing, Q., and Hong-yuan, W., "An Improved Service Scheme Combinated SWAN with QoS-MSR in Ad-hoc Networks," Proc. IEEE 2010 International Conference on Logistics Systems and Intelligent Management, Harbin, pp. 465-468, 2010.

[10] Magula, P., and Kotočová, M., "Quality of Service Models for Mobile Ad-hoc Networks," Proc. ACM 2011 Conference on Computer Systems and Technologies, Vienna, Austria, pp. 448-453, 2011.

[11] Juan, R., Lloret, J., Jimenez, J., and Sendra, S., "MWAHCA: A Multimedia Wireless Ad Hoc Cluster Architecture," The Scientific World Journal, pp. 1-14, 2014.

[12] Bouhouche, H., and Fatmi, S., "A QoS-based Resources Reservation Mechanism for Ad hoc Networks," International Journal of Computer Applications, pp. 29-34, 2010.

[13] Khoukhi, L., Masri, A., and Gaiti, D., "A Hybrid Stateless QoS Approach for Wireless Mobile Ad hoc Networks," Proc. IEEE 2011 Conference on New Technologies, Mobility and Security, Paris, pp. 1-5, 2011.

[14] Helmle, S., Dehm, M., Kuhn, M., Lieckfeldt, D., and Pesch, D., "A Simplified Node Selection Algorithm for Multicast Resource Reservation in TDMA-based Narrowband Mobile Ad-hoc Networks," Proc. IEEE 2013 Wireless Days (WD), Valencia, pp. 1-7, 2013.

[15] Kamruzzaman, S., and AlGhamdi, A."A Resource Reservation Scheme in Cognitive Radio Ad Hoc Networks for C4I Systems," Proc. IEEE Conference on Advanced Information Networking and Applications Workshops, Victoria, BC, pp. 879 – 886, 2014.

[16] Keming, D., and Yahui, Y., "A QoS Routing for Maximum Bandwidth in Ad Hoc Networks," Proc. IEEE Second International Conference on Future Networks, Sanya, Hainan, pp. 343-345, 2010.
[17] Cláudia, A., Luis, J., and Garcí-a, V., "A Distributed QoS Mechanism for Ad hoc Network," International Journal of Ad Hoc and Ubiquitous Computing, pp.25-33, 2012
[18] Vijayalakshmi, A., and Ramamoorthy, P., "Performance Analysis of Genetic Algorithm Based Unicast and Multicast Routing Optimization Model for Mobile Ad hoc Networks," Journal of Theoretical and Applied Inf. Technology, pp. 183-192,2013.
[19] Zhang, X., Zhang , Y., Yan, F., and Vasilakos, A., Interference-Based Topology Control Algorithm for Delay-Constrained Mobile Ad Hoc Networks," IEEE Transaction on Mobile Computing, pp. 742-754, 2014.
[20] Seth, D., Patnaik, S., and Pal, S., "MFQMAC- A Faired and QoS Assured MAC protocol for Multihop Adhoc Network," Journal of Computer Applications, pp. 28-34, 2013.
[21] Ghazali, O., and Harun, N., "Implementation and Validation of an Adaptive FEC Mechanism for Video Transmission," Journal of Information and Communication Technology, pp. 55-74, 2012.
[22] Perkins, C., Belding-Royer, E. and Das, S., "Ad hoc On-Demand Distance Vector (AODV) Routing," US: Internet Engineering Task Force (IETF), 2003.
[23] Sun, Y., Gao, X., Belding-Royer, E., and Kempf, J., "Model Based Resource Prediction for Multi-hop Wireless Networks," Proc.IEEE 2004 International Conference on Mobile Ad hoc and Sensor Systems, Florida, pp. 114-123, 2004.
[24] IEEE Computer Society., "IEEE Standard for Wireless LAN Medium Access Control and Physical Layer Specification", 1999.
[25] Floyd, S., and Jacobson, V., "Random Early Detection Gateways for Congestion Avoidance." IEEE-ACM Transation on Networking pp. 397-413, 1993.
[26] Kevin, F., and Kannan, V., "NS Notes and Documentation. UC Berkeley, LBL, USC/ISI, and Xerox PARC: The VINT Project", 2011.