Route Recovery Algorithm for QoS-Aware Routing in MANETs

Wilder Castellanos, Pau Arce, Patricia Acelas, and Juan C. Guerri

Multimedia Communication Group, ITEAM Institute,
Universitat Politècnica de València
{wilcashe,patacdel,paarvi}@iteam.upv.es, jcguerri@dcom.upv.es

Abstract. The communications inside MANETs usually show frequent disruptions due to changes in topology and the condition of having a shared physical channel. It is necessary to implement a mechanism to maintain connectivity and ensure Quality of Service (QoS). In this paper we introduce a route recovery mechanism for a QoS routing protocol called AQA-AODV (Adaptive QoS-Aware AODV) which is an extension of the AODV protocol. Our proposal provides a mechanism to detect the link failures in a route and reestablish the connections taking into account the conditions of QoS that have been established during the route discovery phase. The simulation results reveal performance improvements in terms of packet delay, number of link failures and connection setup latency while the end-to-end throughput is not affected compared with the throughput achieved by other protocols like AODV.

Keywords: Wireless ad hoc networks, quality of service-aware routing, route recovery, link failure.

1 Introduction

A mobile ad hoc network (*MANET*) is a group of autonomous wireless devices organized themselves dynamically in a mesh topology. The key feature of this type of networking is the nonexistence of any permanent infrastructure. Due to these infrastructure-less and self-organized characteristics, *MANET* encounters different problems from infrastructure-based wired network, such as bandwidth-constrained, variable capacity links and energy-constrained operation. Moreover, routes may include multiple hops because communications need to use intermediate nodes as routers in order to communicate with nodes that are out of its transmission range. This mobility of nodes causes frequent link failures and high error rates, so it makes difficult to maintain the desired QoS in the network. Additionally, due to the fact that the wireless channel is shared among neighbor hosts and that network topology can change as hosts move, the transmission of time-sensitive data (e.g. video packets) is made more difficult. Especially in applications that generate a huge data volume that is delay-sensitive and bursty, since losses of some important data segments (such as synchronization data) may seriously disrupt a long sequence of frames [1].

J. Del Ser et al. (Eds.): MOBILIGHT 2011, LNICST 81, pp. 81–93, 2012.

The main issue is how to efficiently transmit a large volume of delay-sensitive data when many packets are dropped due to the fact that network resources are limited and time-varying. We propose in [2] a QoS-aware routing protocol (*AQA-AODV, Adaptive QoS-Aware AODV*) which is a modified and enhanced version of the Ad hoc On-demand Distance Vector (*AODV*) [3] that allows the source to adapt the transmission rate. More precisely, we have introduced into the original AODV protocol, an adaptive feedback scheme and two mechanisms: one for the estimation of the available bandwidth in the node and the other for the prediction of the consumed bandwidth for a route of multihops. In addition, a QoS extension is added to the *AODV* control packets and the routing table. The result is a QoS path finding mechanism that can provide feedback to the application about the current network state in order to allow the application to appropriately adjust the transmission rate.

To support QoS routing on MANETs, new protocols, like *AQA-AODV*, need to use an efficient route maintenance mechanism. In this paper, we propose a route recovery mechanism for *AQA-AODV*, which not only has to re-establish the connections but it also has to take into account the conditions of QoS that have been established during the route discovery phase.

In order to test the performance of our route recovery model, we have implemented the proposed solution in the ns-2 simulator [4]. Results indicate that the packet delay, link failures and the connection setup latency decrease significantly while the overall end-to-end throughput is not impacted.

This paper is organized as follows. Section 2 briefly reviews some related works. Section 3 describes the main components of *AQA-AODV* protocol. Section 4 presents the performance evaluation of our route recovery mechanism and Section 5 offers some conclusions.

2 Related Works

Several approaches have been proposed based on AODV routing protocol. In [5], Pan *et al.* suggest the approach with two routing protocols called AODV - Local Repair TTL (*AODV-LRT*) and AODV - Local Repair Quota (*AODV-LRQ*). In these approaches are decreased the breadth and depth, of route repair requests. Decreasing the breadth of the repair mechanisms means to limit the maximum number of hops that *RREQ* packets have to pass, assuming that the size of the network topology and transmission range of every node are known. Decreasing the depth of the repair mechanisms means to limit the number of times a node is allowed to forward the route repair request. In [6], Youn et al. propose a new local repair scheme using promiscuous mode which is mainly composed of two parts: adaptive promiscuous mode and quick local repair scheme. Adaptive promiscuous mode repeats the switching processes between promiscuous mode and non-promiscuous mode. The proposed scheme adopts promiscuous mode such that each node keeps monitoring the overheard packets from which the routing information about the route path in adjacent nodes can be obtained. This action can cause excessive energy consumption and reduce network efficiency.

The solutions proposed by Pan *et al.* [5] and Youn *et al.*[6] aim to change the *AODV* protocol to make it more efficient in relation to route maintenance, but do not take into account the conditions of QoS, since these were designed for a routing protocol without QoS support, like *AODV*. Other proposed studies can be consulted in reference [7] which offers a survey of AODV-based approaches.

Sarma *et al.* [8] proposed two route maintenance mechanisms for *AODV* with QoS support. One is based on a special local route repair by limiting route recovery flooding to one hop neighbors only. Other one is route recovery by the destination node itself. However these mechanisms fail when try to re-establish connection to destination with the QoS conditions that had been negotiated during the initial route discovery phase. Other QoS routing protocols for *MANETs* with route recovery mechanisms are described in references [9], [10] and [11]. They are based on AODV with QoS extensions using a model of admission control according to *QAODV* internet draft [12]. However, these solutions do not integrate an adaptive feedback scheme by which the source node can easily adapt its transmission rate according to the state of the route.

3 AQA-AODV: QoS Routing Protocol with Adaptive Feedback Scheme for MANETs

The AQA-AODV (*Adaptive QoS-Aware Ad-hoc On-demand Distance Vector*) protocol, is a QoS routing protocol based on *AODV*, designed with the following modifications:

1. New fields in the packets used in the route discovery phase (*RREQ*, Route Request and *RREP*, Route Reply) to the bandwidth requirements and a "session ID", used to identify each QoS flow that is established.
2. An intermediate node receiving *RREQ/RREP* packets with QoS extension must examine whether it can satisfy the QoS requirements or not in order to rebroadcast/forward the packet to the next hop
3. Algorithms used for the estimation of the available bandwidth that allow nodes along the path to know their available resources (in terms of bandwidth).
4. An adaptive feedback scheme by which the source node can easily adapt its transmission rate according to the state of the route.

3.1 Route Discovery in AQA-AODV

If a source node requires a route to a destination node with specific bandwidth requirements, it broadcasts a *RREQ* packet with the QoS extension (*QRREQ*) to its neighbor nodes. When a node receives a *QRREQ* packet, a reverse route entry is created with the *session ID*, and the *QRREQ* packet is rebroadcasted as in *AODV*. This process continues until the *QRREQ* packet reaches the destination node. In *AODV*, when a destination node or an intermediate node has a "fresh enough" route to the destination, it sends a route reply message to the source [3]. However, only the destination will be able to send the route reply packet (*QRREP*) in *AQA-AODV*. This will ensure that all nodes in the selected route satisfy the bandwidth constraints. When the destination node receives a *QRREQ* packet, if it is a new request, a reverse route entry for the new session is created. Before sending the *QRREP* to the source, local available bandwidth is checked. However, it is not enough to affirm that the route can offer the required bandwidth indicated in the *QRREQ*. The reason is the mutual interference between packets of the same flow, also called "Intraflow contention" [13]. Therefore, one final check is necessary in the destination node. To estimate the intraflow contention, we use the relation between the number of hops and the end-toend throughput. Since the destination node is the last host, it can determine its distance from the source (by the number of hops in *QRREQ*). This information will allow the node to estimate the bandwidth along a path taken into account the contention between packets of the same flow. Figure 1a shows the host's working procedure after receiving a *QRREQ*. Finally, the *QRREP* will be transmitted to the source with a modified header that includes the minimum value between required bandwidth for the source and the maximum bandwidth that all hosts along the route could support taken into account the intraflow contention. Once an intermediate node receives the *QRREP* packet, it compares its available bandwidth with the bandwidth indicated in the *QRREP*. If its local available bandwidth is lower, it updates the min-bandwidth field in *QRREP*, using its available bandwidth. Otherwise, the node forwards the *QRREP*. This procedure will ensure that the source knows the minimum bandwidth along the path which will be the maximum rate that it may transmit. The procedure is shown in Figure 1b. Figure 2 illustrates the overall operation of the key phases of *AQA-AODV*.

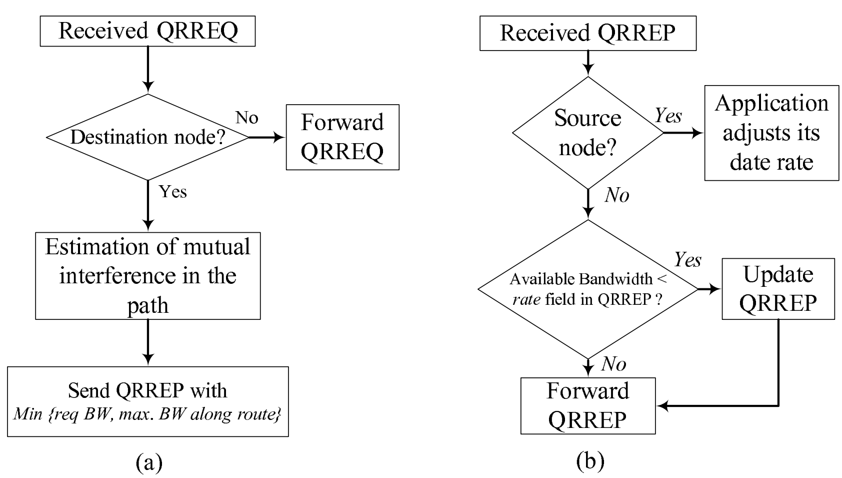


Fig. 1. Procedure in nodes after receiving a QRREQ (a) and QRREP (b)

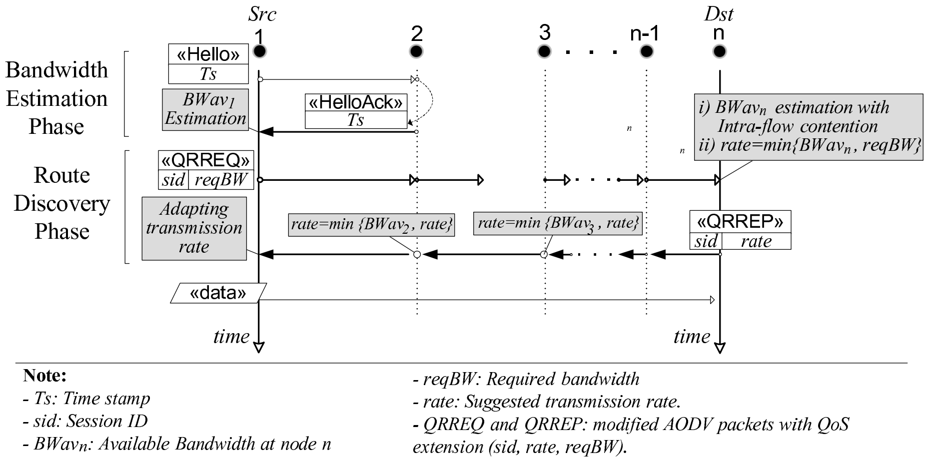


Fig. 2. Overview of AQA-AODV

3.2 Route Recovery Mechanisms for AQA-AODV

Due to changes in topology because of the mobility of the nodes and the condition of having a shared physical channel, the communications inside *MANETs* usually show frequent disruptions. For this reason, it is necessary to implement a route recovery mechanism. This mechanism not only has to re-establish the connections but also take into account the conditions of QoS that have been established during the route discovery phase.

The implemented route discovery mechanism in *AQA-AODV* detects the connection losses in a route when a host doesn't receive a *Hello* message from a neighbor during an interval of time. The *Hello* messages may not be received for three main reasons:

Case 1. There is total connectivity but due to congestion some of the *Hello* messages are lost.

Case 2. The neighbor node is no longer available because it is out of transmission range and the node should look for a new path to the destination.

Case 3. The node is no longer available in the ad hoc network.

Our route recovery mechanism implemented in *AQA-AODV*, perfectly works in any of the two previous cases in which connection recovery is possible. To explain the functionality of the proposed route recovery mechanism in detail, we show two examples for case 1 and case 2. The example showed in Figure 3 consists of a network with four nodes in which each node is inside the transmission range of its one hop neighbors and inside the interference range of its two hop neighbors. Node 1(source node) broadcast a *QRREQ* message to obtain a route to node 4 with a transmission rate of 1 Mbps. When the destination node checks that the maximum transmission rate available is 0.5 Mbps, it sends the information to the source using a *QRREP* message. During the backward process of the *QRREP*, each node checks its own available bandwidth and compares it to the value included in the *QRREP* message. This process has been described in previous sections and we will refer to it as standard procedure of route discovery. Moreover, each node adds a register in the session cache list associated to a session identifier (sid) and an expiration time (*Expiration Time*) with the aim of erasing the old registers (see Figure 3a). Once the route from node 1 to node 4 is defined, data packets are sent through the network. Each time a node gets a data packet related to that session, it updates the expiration time of the registers, avoiding the elimination of the register and keeping the session alive. When some of the hello messages sent by node 4 are lost due to congestion, node 3 detects a link failure and it sends an error message (*RERR*) to the source, including the affected session identifier (*esid*) (See Figure 3b).

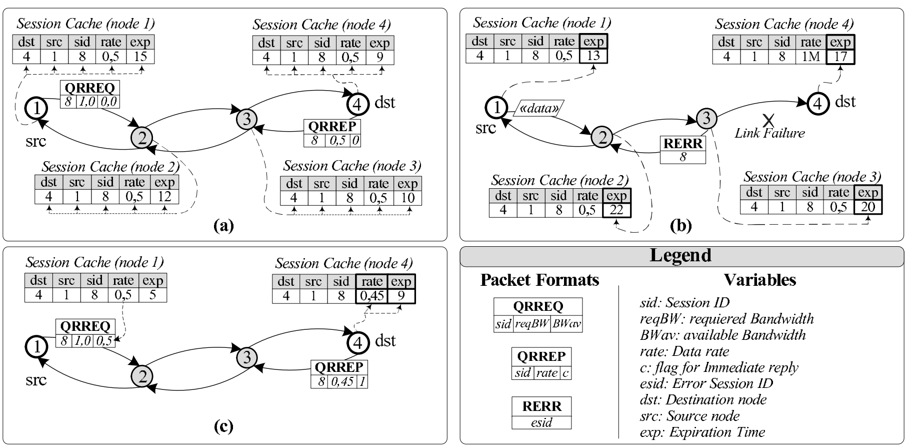


Fig. 3. Example of route recovery mechanism. Case 1.

When node 1 receives the *RERR* message, it queries its session cache list using the session identifier received in the *RERR* message (esid, Error Session ID). Therefore the source sends a *QRREQ* message which includes the required bandwidth, the actual data rate and the session identifier (see Figure 3c).

When the destination node receives the *QRREQ* message it checks if it has a register with the same sid as the one sent by the source in the *QRREQ*. If it does have one, the destination creates a *QRREP* message with the same session identifier, the supported maximum data rate – this rate may be different from the original one – and an immediate reply flag (c = *1 immediate reply*, c = *0 standard reply*). The immediate reply flag warms the intermediate nodes not to execute the standard procedure to verify the available bandwidth but send the *QRREP* message directly to the next hop back to the source. As a conclusion, the route recovery mechanism tries to re-establishconnection to destination with the QoS conditions that had been negotiated during the initial route discovery phase.

In Figure 4, we have the same conditions as in the previous example. However, a new node (node 5) has been added and it does not take part in the present route between nodes 1 and 4. In Figure 4a, we can see the information of the previous session established in the ad hoc network, using the procedure mentioned before. We also suppose that node 4 is moving in the opposite direction of node 3 and, at that moment, there will be a link failure.

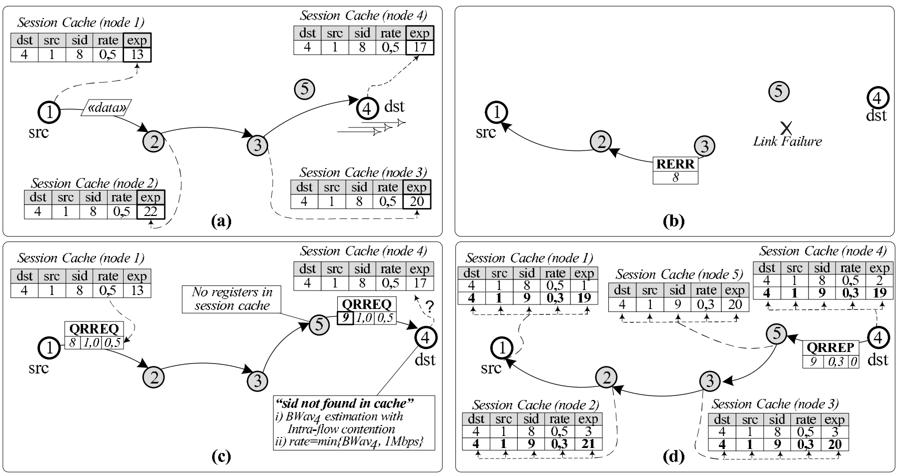


Fig. 4. Example of route recovery mechanism. Case 2.

The link failure between nodes 3 and 4 will be detected in a similar way as it was shown in Figure 3 using a *RERR* message sent to the source (Figure 4b). Nevertheless, now it is possible to achieve the destination node (node 4) through node 5, which is inside the transmission range of nodes 3 and 4. When node 1 receives the error message, it sends a *QRREQ* message and node 5, after processing the message, without finding a register associated to a session identifier (*sid*), proceeds to generate a new *sid* (Figure 4c). This makes the difference with the example shown in Figure 3. For this reason, node 4 does not take into consideration the information of the previous session and it analyses the route request in the standard way. Therefore, it calculates the available bandwidth again and compares it with the bandwidth requested by the source (1 Mbps). The response to the route request is sent to the source through the intermediate nodes using the standard way. These nodes create a new register in session cache and check if they have enough bandwidth to transmit the traffic (Figure 4c). Once the source receives the *QRREP* message, it adapts its transmission rate according to the available bandwidth calculated in the route recovery mechanism, and it starts sending packets to the destination. Registers in the session cache in each node are erased when the time-out expires.

4 Performance Evaluation

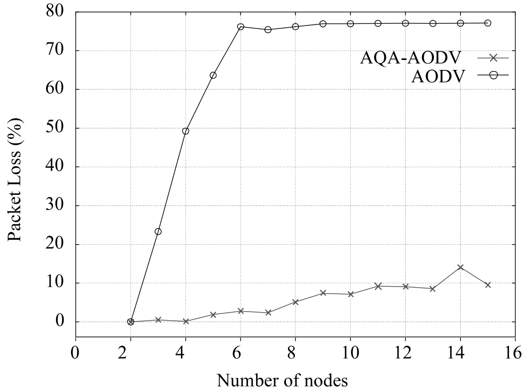
The performance of the proposed route recovery algorithm was evaluated using Network Simulator (*NS-2*). This simulator implements the IEEE802.11 protocol for the MAC layer, working in the Distributed Coordination Function (DCF) mode with a channel data rate of 2 Mbps. The radio propagation model is Two Ray Ground and queue type is Drop Tail with maximum length of 50. The transmission range and interference range are 250 m and 550 m respectively. The performance of our route recovery mechanism was evaluated by comparing it with conventional AODV protocol, using two simulation scenarios: the first scenario consists of a static linear topology with variable length where the parameters was evaluated as a function of the chain length and the second scenario consists of 30 mobile nodes in a rectangular field, 1000m x 1000m, and the mobility model uses the random waypoint model.

4.1 Simulations Results

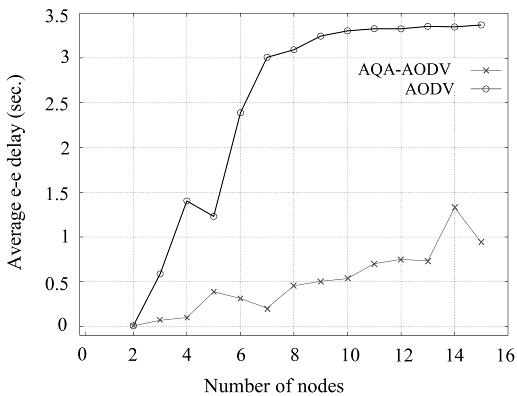
Scenario 1: Static linear topology with variable length. The first scenario consists of a chain of nodes where the performance was evaluated as a function of the chain length. In this scenario, the performance of *AQA-AODV* is tested as function of the number of hops on the path. Node 1 is the source of data traffic and the last node in the chain is the traffic sink. Initially the source required a transmission rate of 0,9 Mbps which be maintained constant when *AODV* is used, but can be changed by the source when *AQA-AODV* is used in the network due to the adaptive feedback scheme.

As seen in Figure 5, using *AQA-AODV*, the network congestion is significantly reduced. Therefore, the time used for waiting in the packet queue and contending for the channel decreases. In other words, our adaptive feedback scheme and our route recovery mechanism allow getting an important decrease in packet loss (Figure 5a) and delay (Figure 5b) without any bandwidth sacrifice.

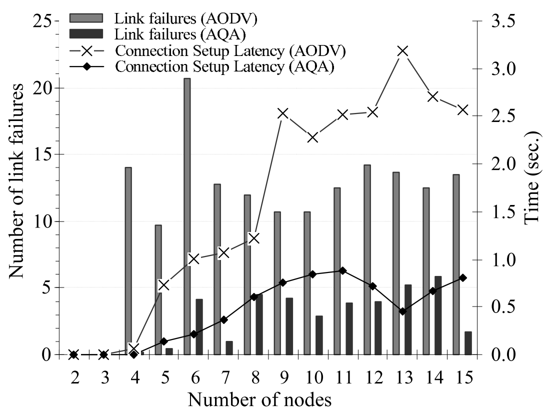
Figure 5c shows the number of link failures and the Connection Setup Latency (*CSL*). *CSL* is the latency incurred in establishing new connection from source to destination after the previous connection is lost (which includes route break detection time and recovery time). In Figure 5c, we notice that, when the chain has 3 or more nodes, the transmission rate (0.9Mbps) is not supported efficiently and the number of link failures drastically increases for *AODV*, which is about 25% higher than link failures for *AQA-AODV*. CSL (*Connection Setup Latency*) in case of *AQA-AODV* is the lowest and varies from 0.05s (for a chain of 4 nodes) to 0.5s (for a chain of 15 nodes) due to reduced recovery time in comparison with *AODV*.



(a)

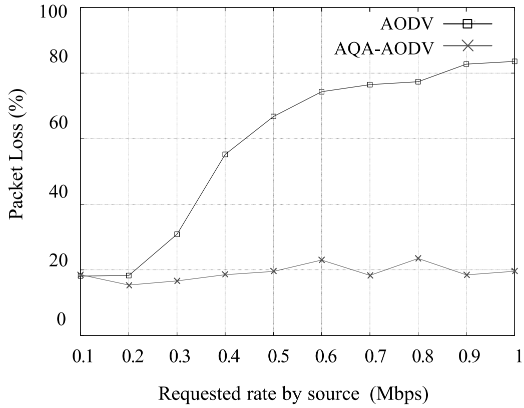


(b)

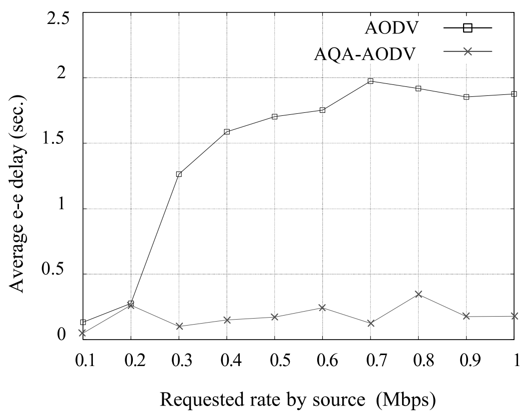


(c)

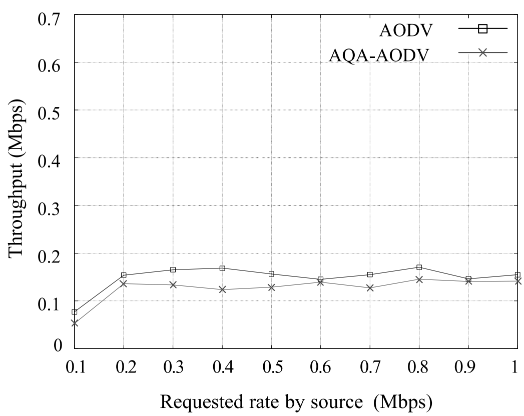
Fig. 5. Packet Loss (a), Average end-to-end delay (b) and Number of link failures and CSL, Connection Setup Latency (c) with variable chain length



(a)



(b)



(c)

Fig. 6. Packet Loss (c), Average end-to-end delay (b) and Throughput (c) with variable requested rate

Scenario 2: Mobile Topology. The second scenario consists of 30 nodes move in a 1000m x 1000m area according to the random waypoint model with pause time set to 20sec. The nodes move toward a random destination using a speed between 0 – 3 m/s. A random source-destination pair sends packets using a request rate between 0,1 and 1,0 Mbps. All traffic flows are Constant Bit Rate (*CBR*) streams over UDP with a packet size of 1000 bytes. Figure 6 shows the results of our simulations in which the packet loss, average end to end delay and throughput are plotted versus the requested rate by source node. In terms of packet loss (Figure 6a), *AQA-AODV* shows great improvement over *AODV*, which achieves very high packet losses for some requested rates. For example, the packet loss is between 19% and 83% using *AODV*, whereas using *AQA-AODV* the packet loss remains lower than 24%.

Figure 6b shows that the average end to end delay of *AQA-AODV* is always below 0,4s, whereas, the end to end delay of *AODV* increases badly when the transmission rate increases from 200 kbps to 1000 kbps. With *AODV*, the maximum average end to end delay reaches 1,9s at 700 kbps, about 16 times higher than using *AQA-AODV*. As seen in Figure 6c the total network throughput achieved with *AQA-AODV* is very close to throughput achieved using *AODV*. We would expect the *AQA-AODV* protocol's performance will degrade in scenarios with high mobility because the nodes will need a specific time for exchanging information about the network status. We observe in figure 7 that in a scenario with mobile nodes the frequencies of route break increase in the network compared with previous static scenario where the link failures were caused by congestion in the nodes. Each time a route breaks due to node mobility, there is some latency in new connection setup (which includes route break detection time, route discovery time and recovery time) and packet gets lost during

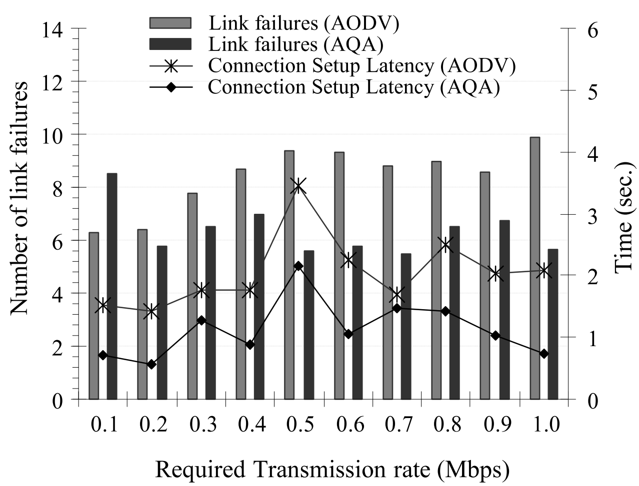


Fig. 7. Number of link failures and CSL (Connetion Setup Latency) with variable requested rate

connection setup period which could explain the growth of the packet loss. Figure 7 shows that the *CSL* of *AQA-AODV* is always lower than *CSL* of *AODV* (about 50% of average lower than *CSL* for *AODV*).

5 Conclusions

The proposed route recovery algorithm for QoS routing protocol (*AQA-AODV*) can contribute to the diminishing of the latency incurred in establishing new connection from source to destination after the previous connection is lost. Our approach incorporates new fields in the route request and route reply packets, an extension of the routing table and the implementation of a session cache table where registers of the active sessions are stored. Simulations show that our proposed mechanism is perfectly integrated into adaptive feedback scheme of *AQA-AODV*. This reduces significantly the dropping rate, the end-to-end delay and the connection setup latency, without impacting the overall end-to-end throughput. In the future, we plan to examine how to implement a hybrid algorithm using source and local repair, which would decrease the connection setup latency and improve the performance in mobile environments.

Our main goal is to implement a framework where the video source exploits the feedback information from the underlying protocol (*AQA-AODV*) to tune a parameter on the source coding in order to adapt the traffic rate to the path. Moreover, this framework must include the route recovery mechanism that would allow nodes to repair link failures with previous QoS conditions.

References

1. Wu, D., Hou, T., Zhu, W., Lee, H.-J., Chiang, T., Zhang, Y.-K., Chao, H.J.: On End-to- End Architecture for Transporting MPEG-4 Video Over the Internet. IEEE Transactions on Circuits and Systems for Video Technology 10(6), 923–941 (2000)
2. Castellanos, W., Acelas, P., Arce, P., Guerri, J.: Evaluation of a QoS-Aware Protocol with Adaptive Feedback Scheme for Mobile Ad Hoc Networks. In: 6th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, QShine 2009, Las Palmas, Gran Canaria (2009)
3. Perkins, C., Royer, E.M., Das, S.: Ad hoc on-demand distance vector (AODV) routing. IETF RFC 3561 (2003)
4. The Network Simulator NS-2, `http://www.isi.edu/nsnam/ns/`
5. Pan, M., Chuang, S., Wang, S.: Local repair mechanisms for on-demand routing in mobile ad hoc networks. In: Proceedings of Pacific Rim International Symposium on Dependable Computing (2005)
6. Youn, J., Lee, J., Sung, D., Kang, C.: Quick local repair scheme using adaptive promiscuous mode in mobile ad hoc networks. Journal of Networks 1(1), 1–11 (2006)
7. Pereira, N., de Moraes, R.M.: A comparative analysis of AODV route recovery mechanisms in wireless Ad Hoc networks. In: IEEE LATINCOM 2009 Conference on Communications, pp. 1–6 (2009)

8. Sarma, N., Nandi, S., Tripathi, R.: Enhancing Route Recovery for QAODV Routing in Mobile Ad Hoc Network. In: The International Symposium on Parallel Architectures, Algorithms, and Networks (2008), doi:10.1109/I-SPAN
9. De Renesse, R., Friderikos, V., Aghvami, H.: Towards Providing Adaptive Quality of Service in Mobile Ad-Hoc Networks. In: IEEE VTC, Melbourne (2006)
10. Zhang, Y., Gulliver, T.: Quality of service for ad hoc on-demand distance vector routing. In: Proceedings of WiMob 2005, vol. 3, pp. 192–196 (2005)
11. Kumar, V., Ilanchezhiapandian, G.: Enhancement of AODV Routing protocol to provide QOS for MANET. In: Proceedings of the National Conference on Emerging Trends in Computing Science NCETCS (2010)
12. Perkins, C., Royer, E.M.: Quality of service for ad hoc on-demand distance vector routing. IETF Draft (2004), `http://www.psg.com/~charliep/txt/aodvid/qos.txt`
13. Sanzgiri, K., Chakeres, I., Belding-Royer, E.: Determining intra-flow contention along multihop paths in wireless networks. In: Proc. Broadnets 2004 Wireless Netw. Symp., pp. 611–620 (2004)