



**HAL**  
open science

## Latency filtering for Q-routing on wireless networks

Alexis Bitailou, Benoît Parrein, Guillaume Andrieux, Killian Couty

► **To cite this version:**

Alexis Bitailou, Benoît Parrein, Guillaume Andrieux, Killian Couty. Latency filtering for Q-routing on wireless networks. 17th Int. Wireless Communications & Mobile Computing Conference - IWCMC 2021, Jun 2021, Harbin (virtual), China. 10.1109/IWCMC51323.2021.9498737 . hal-03279776

**HAL Id: hal-03279776**

**<https://hal.science/hal-03279776v1>**

Submitted on 23 Aug 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Latency filtering for Q-routing on wireless networks

Alexis Bitaille

University of Nantes

LS2N Lab, Nantes, France  
alexis.bitaille@ls2n.fr

Benoît Parrein

University of Nantes

LS2N Lab, Nantes, France  
benoit.parrein@ls2n.fr

Guillaume Andrieux

University of Nantes

IETR Lab, Nantes, France  
guillaume.andrieux@univ-nantes.fr

Killian Couty

University of Nantes

LS2N Lab, Nantes, France  
killian.couty@etu.univ-nantes.fr

**Abstract**—Q-routing is inspired by Q-learning, a reinforcement learning algorithm. Originally, it uses latency as routing metric. But, latency can be difficult to estimate especially in a noisy wireless environment. In this paper, we propose to filter the latency measure with a moving average, in order to improve the quality of service metrics such as packet delivery ratio and average delay. We compare our modification to the original Q-routing and use OLSRv2 as reference routing protocol. We observe an improvement of the average delivery time on a wireless grid of 3 % compared to the original Q-routing. On our mobility scenario, the number of routes changes is at least twice lower (from 6 to 3 route changes between the two approaches in this scenario). The gain on QoS metrics depends mainly on the speed of the nodes. These improvements are obtained without making Q-routing more complex as a moving average is added. We provide all the materials to conduct reproducible research on our public git repository.

**Index Terms**—wireless routing, reinforcement learning, Q-routing, Qualnet simulator, quality of service

## I. INTRODUCTION

Latency is considered as not reliable and noisy routing metric [1]. Q-routing [2] is one of these algorithms using directly the latency as routing metric. In their paper, Boyan and Littman [2] shown that Q-routing can offer good performance compared to a hop count-based algorithm. As they evaluated Q-routing on their home simulator, they eluded the problem of the noise of measure of the latency. Even on implementation of Q-routing [3]–[5] on network simulator, the problem wasn't mentioned but real if we take a look to quality of service results such as packet delivery ratio and average delivery time.

In our previous work [6], we implemented Q-routing on the network simulator Qualnet. We shown that the greedy strategy used by Q-routing had a drawback. Q-routing didn't go back the fastest path if there was a brief congestion on it. We proposed a modification of Q-routing to reduce this side effect of the greedy strategy. We evaluated Q-routing and its modification on two different wireless topologies, including the wireless version of the irregular grid of [2]. The simulated scenarios was static. We shown that our proposal reduced the main drawback of the greedy strategy. Moreover, it slightly improved QoS metrics such as packet delivery ratio and average delivery time.

In this paper, we show that the measurement of latency can be noisy. This could lead Q-routing to make decision on noise. So, we propose to integrate to Q-routing a filtering mechanism in order to reduce the noise of measure on the latency, so improve the quality of the Q-values, the routing metric. We

also propose a simple mechanism to reduce the influence of the initialization value. We evaluate our modified Q-routing with filtering on several scenarios on ad-hoc wireless networks, including a mobility scenario. We compare it to the original Q-routing. The implementation nuOLSRv2 of OLSRv2 [7] is also used as reference.

The organization of the paper is the following. In Section II, we summarize some previous works about Q-routing. In Section III, we detail the implementation of our distributed Q-routing protocol. Section IV defines the experimental setup. Section V provides results in terms of QoS and a discussion as well. The last section concludes the work and draws some perspectives.

## II. RELATED WORK

In this section, we see in more details Q-routing algorithms and other related works.

### A. The original Q-routing

Watkins and Dayan [8] created a reinforcement learning algorithm called Q-learning in 1992. Two years later, Boyan and Littman [2] proposed to integrate Q-learning in routing algorithm. They named their algorithm Q-routing in reference to Q-learning. In this algorithm, each node  $x$  looks for the lowest Q-value, defined using the  $Q$  function. The estimated delivery time from node  $x$  to node  $d$  by node  $y$  is noted:  $Q_x(d, y)$ . They define Q-value of function  $Q$  as:

$$\Delta Q_x(d, y) = \eta(q + s + t - Q_x(d, y)) \quad (1)$$

where  $\eta$  is the learning rate (usually 0.5 in [2])  $q$  the unit of time spent in node  $x$ 's queue,  $s$  the unit of time spent during the transmission between  $x$  and  $y$  and  $t$  as

$$t = \min_{z \in \text{neighbour of } y} Q_y(d, z). \quad (2)$$

In this case, the effective delivery time is the reward  $R$  and defined as:  $R = q + s + t$ . At the beginning, the Q-values are initialized with the value 0. Q-routing has a greedy strategy that means it always chooses the lowest value. So the first choice is very important. In order to make the first choice equitably, an exploration phase is needed to discover all the choices. During this phase, the Q-value is not updated.

Several networks topologies are tested in their work including an  $6 \times 6$  irregular grid. The authors argue that only local

information is used to proceed. The presented results of [2] concern only the 6x6 irregular grid. Q-routing is compared to Bellman-Ford's shortest path algorithm. In their works, Q-routing is not always able to find the shortest path under low network load. Nevertheless, the latency is similar to the shortest path in low load condition. Q-routing clearly outperforms the shortest path in high load condition (even if the high load condition is not well-defined in [2]). However, when the traffic load decreases, Q-routing keeps the high load policy. The original approach is thus not adapted to dynamic changes. So, Q-routing is not design for scenarios with mobility.

### B. AQ-routing

As ahead said, Q-routing is a greedy algorithm. The mobility can easily degrade the performances. Serhani *et al.* [9] proposed a modification for Q-routing in order to improve performances in mobility scenario. They named their extension Adaptive Q-routing (AQ-routing). AQ-routing takes several main concepts from OLSR [10] such as HELLO packets but also ETX metric [11]. Unlike the original Q-routing, AQ-routing doesn't use latency as routing metric. It uses a metric based on link stability:

$$Q_{metric_{ij}} = \alpha_{ij} \cdot \varphi(MF_j) + (1 - \alpha_{ij}) \cdot \lambda ETX_{ij} \quad (3)$$

where  $MF$  is the Mobility Factor,  $\alpha$  the learning rate,  $\varphi(MF_j)$  is defined as:  $\varphi(MF_j) = \frac{a}{1 - e^{-\frac{MF_j}{b}}}$ .

In their paper, they compared AQ-routing to OLSR (standard and with ETX metric version) on ns-3. On static test case, AQ-routing offers the best PDR but the worst average delivery time. On mobility test case, AQ-routing provides a stable average delivery time and the best PDR. Start to 4 m/s, the average delivery time is better with AQ-routing than with OLSR ETX. To obtain this performance, Serhani *et al.* have increased the complexity of Q-routing especially the computation of the reward. For example, the exponential function is quite costly in terms of computation.

### C. Reinforcement Learning for LT Optimisation

Q-routing is not designed for wireless sensor networks (WSN). Bouzid *et al.* [12] proposed to adapt Q-routing to WSN context. In order to improve Q-routing, they changed the reward formula. The reward is the energy weighted by the number of hop. They named their modification Reinforcement Learning for LT Optimisation (R2LTO). They compared their algorithm to the original Q-routing and RLBR, another recent specialized algorithm on their home-made simulator. With their simulation, R2LTO improved the lifespan of the network. Other QoS parameters are not presented. Also, as they evaluated only the algorithm, they don't take the routing management overhead into account. The routing management overhead can degrade the performance.

### D. $Q^2$ -routing

The original Q-routing considers only the latency. Q-routing will select a low latency route even if the path loss more

packets. Recently, Hendriks *et al.* [4] proposed an extension of Q-routing considering also the packet delivery ratio and the jitter. Their algorithm is called  $Q^2$ -routing. They adapted the  $Q$  function to include these QoS metrics:

$$Q_x(d, y) = (C_d \times C_j \times C_l)((1 - \alpha)Q_x(d, y) + \alpha r) \quad (4)$$

where  $C$  are coefficient depending on the traffic QoS requirement,  $\alpha$  is the learning rate and  $r$  is  $q + s + t$  in (1).

In their paper, they evaluated  $Q^2$ -routing on a topology composed of 3 paths on ns-3. It compared to an implementation of the original Q-routing and AODV [13]. Packets loss and delay appeared during the simulation on different paths in order to test  $Q^2$ -routing features. According to their results,  $Q^2$ -routing outperforms AODV and Q-routing in most of the test cases in terms of PDR, average delivery time and jitter. However, their scenario is designed to advantage  $Q^2$ -routing as the simulation event can only detect by  $Q^2$ -routing and some of them by Q-routing. Their reward is also more complex than the original Q-routing. They added 3 coefficients  $C$  fixed or computed with QoS metrics depending on the QoS needs.

### E. Conclusion

These three modifications of Q-routing modify the definition of the reward compared to the original Q-routing to obtain better QoS performance. Indeed, AQ-routing and R2LTO don't use the delay in their Q function.  $Q^2$ -routing kept the delay, however other coefficients computed with QoS metrics were added. In [4] and [12], the modified Q-routing offers the best performance compared to the original Q-routing. Finally, the quality of the measured delay was not discussed in any of these propositions. Those solutions were focused on a new  $Q$  function computation instead of explaining why Q-routing have not the expected results with real network conditions. We argue that Q-routing can offer better performance without modifying the  $Q$  function thanks to a better latency estimation.

## III. Q-ROUTING IMPLEMENTATION DETAILS

In this section, we describe our implementation of Q-routing fully distributed and deployable on wired and wireless networks.

### A. Implementation overview

Our implementation of Q-routing is "bare-bones". There is no auxiliary function as we can have in OLSRv2 for example. We define the maximum route length (16 hops), the timeout delay (60 s), the maximum number of routes per packet (32 routes per packets), and the periodic update delay (10 s). Nodes have access to local information only. Additionally, we add the parameter  $\eta$  from Eq. (1) and the exploration phase duration. The routing table has been replaced by the function  $Q$  inspired of Eq. (1). More details of our original implementation are given in [6]. The implementation of Q-routing is available in a public git repository <sup>1</sup>.

<sup>1</sup><https://gitlab.univ-nantes.fr/l2n-rio/qrouting-qualnet>, it assumes a valid Qualnet license

## B. Route lifecycle

In our modified version Q-routing, we modified the route lifecycle. We add two route state : "on trial" and "verified". This modification reduces the effect of the initialization value and the time-out penalty on the Q-values. For example, when the Q-value is initialized to 0, a new route must be selected even if the route is not optimal. When a route is created, the route is considered as "on trial". A route is chosen if there is any verified route. If there are only "on trial" routes, the route is chosen randomly between them. An "on trial" route becomes "verified" when there is at least 3 updates of this route. The reward is updated at each update while the route is on trial. When a route runs out, it loses its verified status.

## C. Filtering

The filtering method is a simple moving average. The average latency is computed with the last latency values. The maximum number of values is fixed. In our experiments, Q-routing can use up to 7 values. The Q-value is computed with this average value instead of the last latency value. These values are stored in memory locally. When the route runs out, the latency values are dropped. So, two following values are spaced out less than the time-out.

## IV. EXPERIMENTAL SETUP

In this section, we describe the complete experimentation set-up and the results of our simulation. We proposed two topologies: a toy example and a more complex topology inspired from Boyan and Littman works [2]. The Table I sums up the different parameters. We benchmark two routing protocols: our implementation of Q-routing, Q-routing with latency filter and nuOLSRv2 (OLSRv2 [7] Niigata University implementation). OLSRv2 is the successor of OLSR, it is standardized routing protocol specialized in mobile ad-hoc networks (MANETs). We use Scalable Qualnet 9.2.1 as network simulator. It is a commercial network simulator derived from GloMoSim. Qualnet supports parallelization in order to speed up simulation on large scale networks [14]. Thirty seeds are used for each combination of parameters.

Feature	Parameter	Value
Network	Link	Wireless IEEE 802.11a link IEEE 802.11e link layer
Node	Number of queues	1 FIFO queue
	Mobility	only on scenarios with mobility
CBR	Message size	512 bytes
Simulation	Duration	5 min (first scenario) variable (mobility scenario) 30 min (wireless grid)
Q-routing	Exploration	15 s
	$\eta$	0,9
	filtering window	7 samples

TABLE I: Simulation parameters

### A. A mobility toy example

We evaluate Q-routing on the scenario with mobility as depicted on Figure 1. The goal of this scenario is to evaluate

Q-routing and its modification on a simple mobility scenario and compare their results to nuOLSRv2. Our test CBR stream is between node 9 and node 10 which are the source and the destination respectively. The CBR source starts sending at 30 s and stop 1 minute before the end of the simulation. The interval between two messages is 4 ms, so a throughput of 1 Mb/s. The node 9 moves from East to West. Its movement is uniform and rectilinear. The node 9 has to travel 1350 m from the starting point up to the final point. So, the duration of the simulation depends on the speed of the node 9. It starts at 35 s and arrives 2 minutes before the end of the simulation. We increase the frequency of the update of Q-routing for this test case. It sends at least update every 5 s instead of 10 s. We also reduce the timeout to 10 s instead of 60 s.

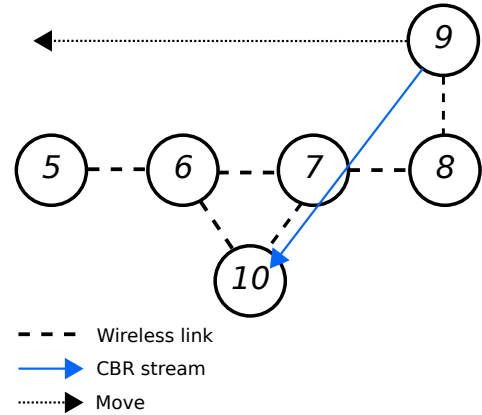


Fig. 1: Mobility toy example topology at  $t = 0$

### B. Wireless irregular grid

We adapted the irregular grid of Boyan and Littman [2] with wireless links. The topology is composed of two grids 4 nodes by 3 linked by two paths. Four nodes (11, 15, 23 and 29) have been removed compared to the wired grid due to their location that created unwanted wireless links. The Figure 2 illustrates this topology in a logical and compact form. For example, the node 9 (on the left block) can only communicate with nodes 3, 10 and 8.

In this case, there are 15 CBR streams on the adapted irregular wireless grid. The location of those CBR streams has been defined randomly. Their starting time and their stop time have been defined randomly but the CBR streams must start after one minute. All CBR streams have the same throughput. The settings of the CBR streams are defined once and don't change between the simulations. The Figure 3 detailed the spatial and temporal location of the CBR stream on the grid. There is 5 CBR streams maximum at the same time. The goal is to evaluate Q-routing and Q-routing with filtering on the adapted wireless grid with non-constant traffic. The simulation time is 30 minutes

## V. SIMULATION RESULTS

In this section, we present the results of the experimentation. We focus on two metrics: the average end-to-end delay (or

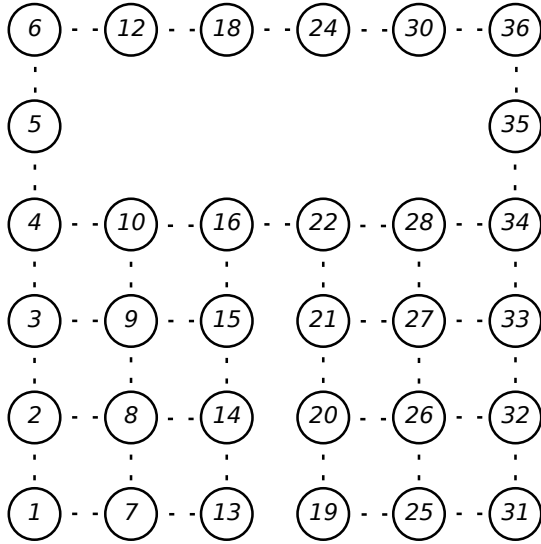


Fig. 2: Wireless irregular grid inspired from Boyan and Littman [2]

average delivery time), the packet delivery rate (PDR). All those metrics are measured at the application layer (layer 7). Disordered messages are dropped by the receiver. Only the messages received and accepted contribute to the average delivery time. The throughput of the CBR streams is expressed at the application layer.

#### A. A mobility toy example

For this scenario, we evaluate Q-routing, its modification with filtering and nuOLSRv2 on a simple mobility scenario.

We focus on the node 9 and the routes to the destination node 10. The Figure 4 shows the Q-values of the current next hop on node 9 over the simulation time for the seed 7 at around 1 m/s. Without filtering, the Q-values are noisy. Q-routing changes of route more frequently or makes not optimal choice. For example, around 550 s, node 9 changes for node 6 and finally goes back to node 7 few seconds later. This case is also present around 850 s. When the route via node 6 is available (around 550 s), the node 9 changes for this route due to the optimistic initialization value. This case also happens with the route via node 5 (around 850 s). When the latency is filtered, the Q-values are less noisy. There are fewer changes of route. Around 550 s, node 9 changes for node 6, but it doesn't go back to node 7 after.

From this example, we compute the number of change of route all the simulations of this scenario. The Figure 5 shows the number of route changes following the speed of the node 9. Q-routing with filtering reduce significantly is the number of route changes. With the original Q-routing, the average number of changes is around 6. With filtering, the average number of change is under 3.

The Figure 6a shows the packet delivery ratio in function of speed of the node 9 for Q-routing and Q-routing with filtering. The packet delivery ratio and average delivery time are very similar between Q-routing and its variant on this test case.

Q-routing with filtering improves slightly the packet delivery ratio start to 4 m/s. The Figure 6b shows the average delivery time in function of speed of the node 9 for Q-routing and Q-routing with filtering and nuOLSRv2. Again, the results between the two Q-routing is very similar. Q-routing with filtering improves slightly the average delivery time from 4 m/s speed.

#### B. Wireless irregular grid

We evaluate Q-routing, Q-routing with filtering and nuOLSRv2 on the wireless grid. The average delivery time and the packet delivery ratio are the average of all the CBR streams. The 15 CBR streams contribute to the average delivery time and the packet delivery ratio. The Figure 7a shows the packet delivery ratio following the throughput per CBR stream for Q-routing, Q-routing with filtering and nuOLSRv2. The three routing protocols have the same shape. The packet delivery ratio decreases when the throughput per CBR increases. Q-routing and Q-routing with filtering offer the same packet delivery ratio. The difference between Q-routing and nuOLSRv2 is limited, around 5 % between Q-routing and nuOLSRv2 in the best case (at 820 kb/s). Overall, Q-routing with filtering delivers less than 1 % more packets than Q-routing and 4 % more packets than nuOLSRv2. The Figure 7b shows the average delivery time in function of the throughput per CBR stream for Q-routing, Q-routing with filtering and nuOLSRv2. As the packet delivery ratio, the curves of the average delivery time have the same shape for the three routing protocols. On average, nuOLSRv2 is the slowest of the three protocols. Q-routing is up to 1 s faster. Q-routing with filtering improves the latency compared to the original Q-routing. It is up to 0.5 s faster. Overall, Q-routing with filtering is 3 % faster than Q-routing and 9 % faster than nuOLSRv2.

#### C. Discussion

The window size for the filtering could be optimized for a specific topology. We choose 7 samples because it gives good results on a not presented test case. It is also a trade-off between the memory space needs and the quality of the filtering. On the mobility toy example, the average number of changes is under 3 (Fig. 5). As the optimal value is 3, the filtering window size is sub-optimal. The effect of the speed on the QoS metrics seems to confirm this hypothesis. So, the QoS metrics can be improved by fine-tuning Q-routing parameters. The moving average could act as a memory and decreases the reactivity of Q-routing. It could be sub-optimal for the wireless grid for example. But, as the test case is static, the memory effect of the moving average is less disturbing. For the filtering, we used a moving average. There are better methods of filtering (e.g. Kalman filters), but they are also more complex.

The wireless technology could also have an impact on the noise. For example, we don't evaluate the impact of the physical and the link layers. IEEE 802.11a with IEEE 802.11e layer could have not the same wireless behaviour that

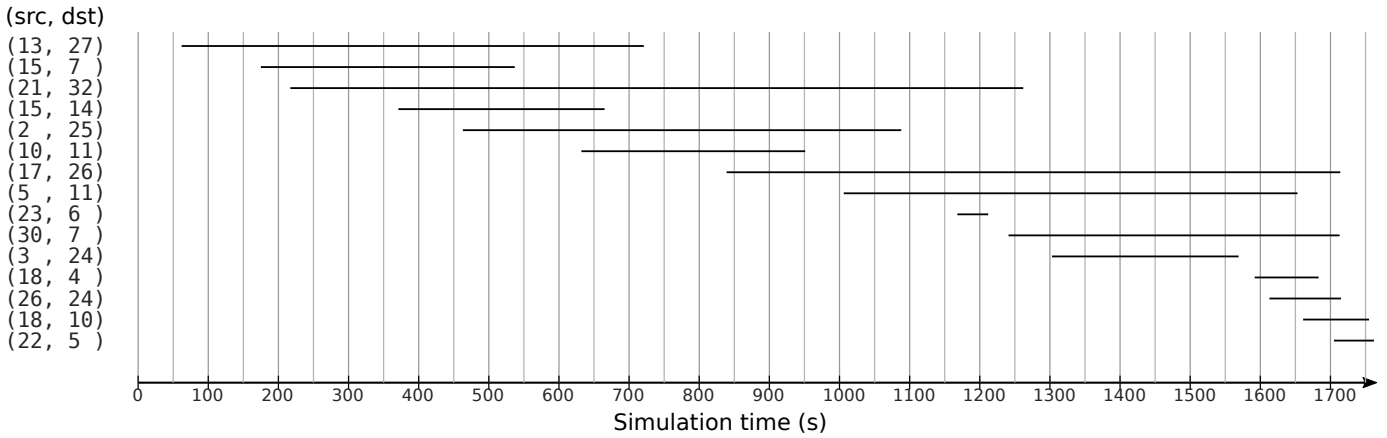


Fig. 3: CBR streams on the wireless grid

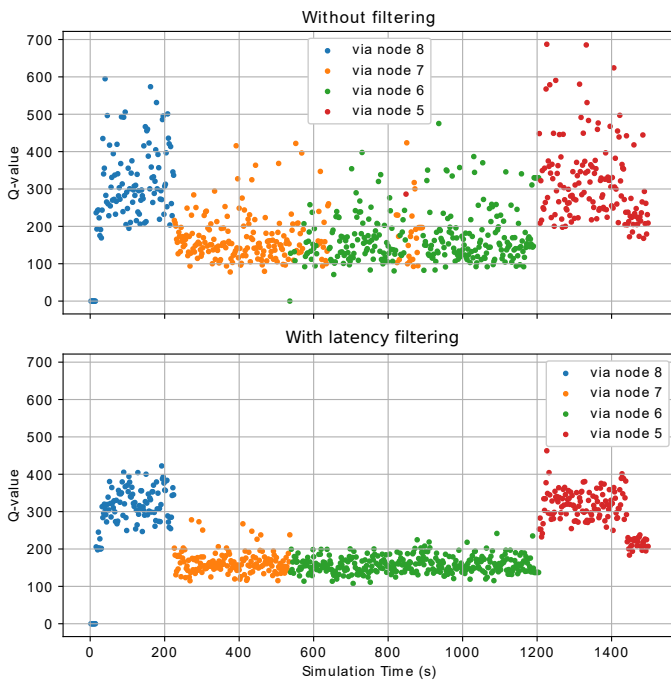


Fig. 4: Q-value by next hop on node 9 over the simulation time on our mobility toy example (seed 7)

IEEE 802.15.4. Also, the influence of other parameters such as interferences are not evaluated.

On the wireless grid, the difference of performance between Q-routing and nuOLSRv2 can be explained by the unnecessary verbosity of nuOLSRv2. In fact, nuOLSRv2 broadcasts more packets than Q-routing. Those packets are useful in mobility scenarios but not in this scenario. Q-routing with filtering delivers the same quantity of packet than the original Q-routing, but faster. As Q-value is less noisy, the routing choices are more stable.

## VI. CONCLUSION

In this paper, we present an evaluation of our modification of Q-routing on the wireless standard IEEE 802.11. We expe-

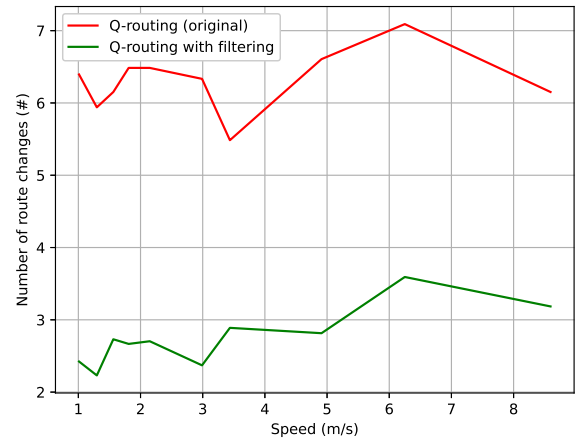
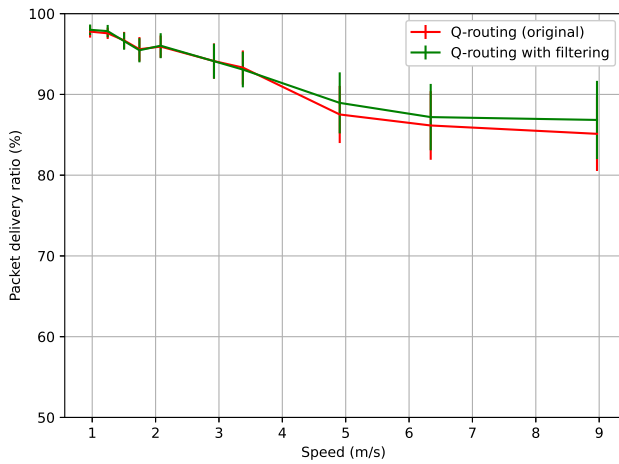


Fig. 5: Average number of route changes on node 9 on our mobility toy example.

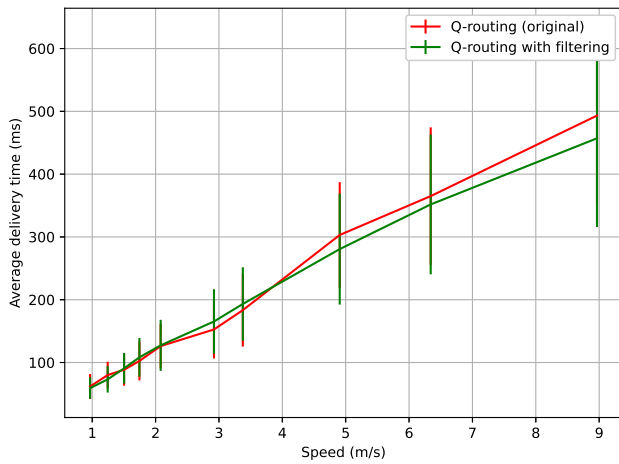
rienced it on the professional packet driven simulator Qualnet on two scenarios. On the wireless grid, the two versions of Q-routing outperform nuOLSRv2 in terms of PDR and latency. Q-routing with filtering improve the latency without degrade the packet delivery ratio compared to the original Q-routing. Even if the filtering doesn't improve so much the PDR and the latency in our mobility scenario, it still has the positive effect by reducing the number of changes of routes. As number of route changes shows, the size of the filtering window can be tuned in order to improve the QoS metrics on a specific test case. We provide all the materials to conduct reproducible research on our public git repository.

## REFERENCES

- [1] R. Baumann, S. Heimlicher, M. Strasser, and A. Weibel, "A survey on routing metrics," *TIK report*, vol. 262, pp. 1–53, 2007.
- [2] J. A. Boyan and M. L. Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach," in *Advances in neural information processing systems*, 1994, pp. 671–678.

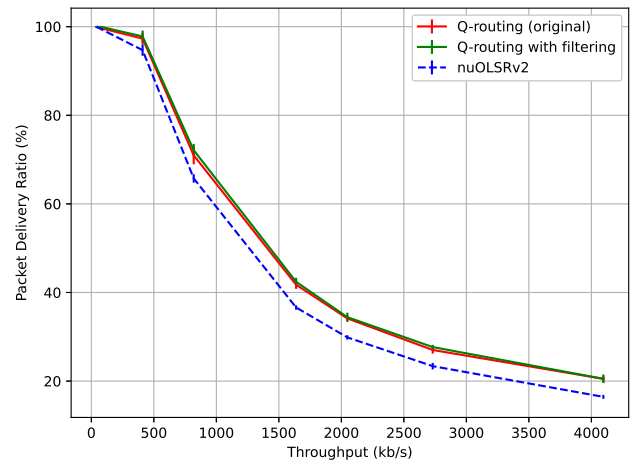


(a) Packet delivery ratio

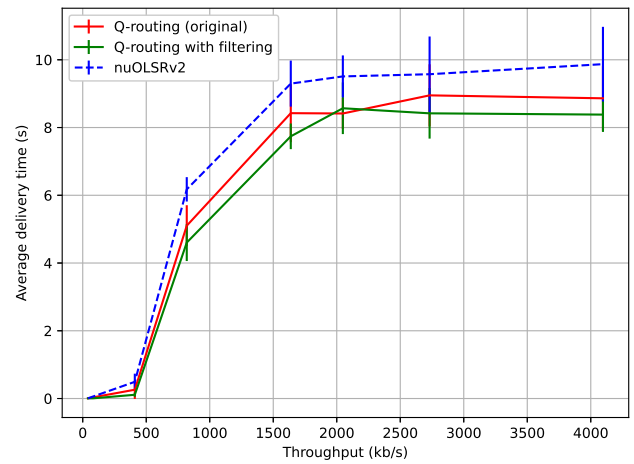


(b) Average delivery time

Fig. 6: Results on our mobility toy example.



(a) Packet delivery ratio



(b) Average delivery time

Fig. 7: Results on the wireless grid scenario.

- [3] S. Hoceini, A. Mellouk, and B. Smail, "Average-Bandwidth Delay Q-Routing Adaptive Algorithm," in *2008 IEEE International Conference on Communications*, May 2008, pp. 1840–1844.
- [4] T. Hendriks, M. Camelo, and S. Latré, "Q2-Routing : A Qos-aware Q-Routing algorithm for Wireless Ad Hoc Networks," in *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct. 2018, pp. 108–115, iSSN: 2160-4886.
- [5] D. K. Sharma, D. Kukreja, P. Aggarwal, M. Kaur, and A. Sachan, "Poisson's probability-based Q-Routing techniques for message forwarding in opportunistic networks," *International Journal of Communication Systems*, vol. 31, no. 11, 2018. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.3593>
- [6] A. Bitailou, B. Parrein, and G. Andrieux, "New Results on Q-Routing Protocol for Wireless Networks," in *Ad Hoc Networks*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, L. Foschini and M. El Kamili, Eds. Cham: Springer International Publishing, 2021, pp. 29–43.
- [7] T. H. Clausen, C. Dearlove, P. Jacquet, and U. Herberg, "The Optimized Link State Routing Protocol Version 2," RFC 7181, Apr. 2014. [Online]. Available: <https://rfc-editor.org/rfc/rfc7181.txt>
- [8] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992.
- [9] A. Serhani, N. Naja, and A. Jamali, "AQ-Routing: mobility-, stability-aware adaptive routing protocol for data routing in MANET-IoT systems," *Cluster Computing*, vol. 23, no. 1, pp. 13–27, Mar. 2020. [Online]. Available: <https://doi.org/10.1007/s10586-019-02937-x>
- [10] T. H. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," RFC 3626, Oct. 2003. [Online]. Available: <https://rfc-editor.org/rfc/rfc3626.txt>
- [11] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, ser. MobiCom '03. San Diego, CA, USA: Association for Computing Machinery, Sep. 2003, pp. 134–146. [Online]. Available: <https://doi.org/10.1145/938985.939000>
- [12] S. E. Bouzid, Y. Serrestou, K. Raouf, and M. N. Omri, "Efficient Routing Protocol for Wireless Sensor Network based on Reinforcement Learning," in *2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, Sep. 2020, pp. 1–5, iSSN: 2687-878X.
- [13] S. R. Das, C. E. Perkins, and E. M. Belding-Royer, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561, Jul. 2003. [Online]. Available: <https://rfc-editor.org/rfc/rfc3561.txt>
- [14] R. Sharma, V. Vashisht, and U. Singh, "Modelling and simulation frameworks for wireless sensor networks: a comparative study," *IET Wireless Sensor Systems*, vol. 10, no. 5, pp. 181–197, May 2020, publisher: IET Digital Library. [Online]. Available: <https://digital-library.theiet.org/content/journals/10.1049/iet-wss.2020.0046>