SAPIENZA
UNIVERSITÀ DI ROMA

"Sapienza" University of Rome
Faculty of Information Engineering,
Informatics and Statistics
Department of Computer Science

Course Name

Lecture notes integrated with the book "My book",
Author 1, ...

Author
Simone Bianco

November 29, 2025

Contents

Information and Contacts

Personal notes and summaries collected as part of the *Course Name* course offered by the degree in Computer Science of the University of Rome "La Sapienza".

Further information and notes can be found at the following link:
https://github.com/Exyss/university-notes. Anyone can feel free to report inaccuracies, improvements or requests through the Issue system provided by GitHub itself or by contacting the author privately:

- Email: **bianco.simone@outlook.it**
- LinkedIn: **Simone Bianco**

The notes are constantly being updated, so please check if the changes have already been made in the most recent version.

Suggested prerequisites:

Small list of prerequisites

Licence:

1

Q-routing literature review

TODO intro

In RL based design, the following aspects are addressed:

- identification of the most appropriate states and actions of the agent
- definition of the reward function depending on the metrics to optimize
- identification of environment model when available

Given a target field of application, different design models may be elaborated, which differ in how they address each of the previous aspects.

In the last 25 years many RL-based routing protocol have been proposed, but most of them share the same high-level structure.

In literature, nodes are confused with agents, and in almost all protocols the reward is (at least partially) calculated by a node upon selecting an entire route to use for all packtes to transmit, or just a next hop to transmit the current data packet. Thus, a *node* should be considered to consist of an agent and optional *modules*:

- **local reward** module: it calculates reward based on local view, which reflects the cost of communication as seen by the packet sender
- **remote reward** module: it receives feedback sent by the next hop or by the destinatino node — if local and remote modules are both employed they are combined to form the reward return to the agent
- **link-state information maintenance** module: it collects useful link state information through periodi or on-demand *Hello packets*

Therefore, the neighboring nodes of a node define the environment of the agent representing a node.

TODO parlare dei tipi di reti?

1.1 Q-routing

Boyan and Littman [BL93] were the first to propose a hop-by-bop routing algorithm based on Q-learning, called Q-routing, and the most of exists RL-based routing protocols today are just extensions of this algorithm. The following is the algorithm that defines Q-routing in detail.

1: **function** QROUTING( )
2: Initialize Q_i matrix randomly
3: **while** termination condition holds **do**
4: **if** packet P is ready to be sent to d **then**
5: Determine node $j^* \leftarrow \arg\min_{j \in \mathcal{N}(i)} Q_i(d, j)$
6: Send packet to node j^*
7: Collect estimate $\theta_{j^*}(d)$ from node j^*
8: Update $Q_i(d, j^*) \leftarrow (1 - \alpha) \cdot Q_i(d, j^*) + \alpha \cdot [W_i^q(P) + T_{ij^*} + \theta_{j^*}(d)]$
9: **end if**
10: **end while**
11: **end function**

We will briefly explain the algorithm. First, let's present the notation:

- i is the node that is currently running the algorithm
- P is a packet that node i needs to forward to destination d
- $Q_i(d, j)$ is the *delivery delay* that i estimates it takes, for node j, to deliver the packet P at destination i
- $\mathcal{N}(j)$ is the set of j's neighbors
- $\theta_j(d)$ is j's estimate for the time remaining in the trip to destination d of packet P
- $W_i^q(P)$ is the time spent by packet P in node i's queue
- T_{ij} is the transmission time between nodes i and j

Each entry of the table Q_i is called **Q-value**, and when node i wants to send a packed, it selects the node j^* that minimizes the Q-value. Upon sendin packet P to node j^*, node i receives back from j^* the value

$$\theta_{j^*}(d) = \min_{k \in \mathcal{N}(j^*)} Q_{j^*}(d, k)$$

Then, node i has to update its estimate of $Q_i(d, j^*)$ based on $\theta_{j^*}(d)$, which can be performed utilizing the formula

$$Q(s_t, a_t) = (1 - \alpha) \cdot Q(s_t, a_t) + \alpha \cdot \left[R_{t+1} + \gamma \cdot \max_{a \in A} Q(s_{t+1}, a) \right]$$

by setting

- $R_{t+1} = W_i^q(P) + T_{ij^*}$ since it represents the *link cost*
- $\gamma = 1$

- $\max_{a \in A} Q(s_{t+1}, a) = \min_{k \in \mathcal{N}(j^*)} Q_{j^*}(d, k)$ since the "action to take" corresponds to choosing an neighbor in this context, however we seek to *minimize* the Q-value since the delay is clearly a decreasing metric

Despite the wide adoption of this protocol over the years, Q-routing is still far from perfect and it has its flaws. Some of the problems are inherent problems of Q-learning in general, such as *slow convergence* rate and high *parameter setting sensitivity*, but there are problem that arise from the protocol itself. For instance, to avoid frequen oscillations of the Q-values — in case of sudden variations of traffic in the network — and limit the overhead of the protocol, Nowe, Steenhaut, Fakir, et al. [NSF+98] proposed an extension in which j^* sends an average $\overline{\theta}_{j^*}(d)$ to i only after a certain number of exchanged packets. Another known problem is called **Q-value freshness**: the estimate $\theta_j(d)$ is evaluated upon packet transmission on a route, therefore if a route is not selected during a long period of time, the agent does not have an accurate estimate of the current condition of such route.

1.2 Classification criteria

The authors underline that, to their knowledge, their work is the first in the literature that proposes classification criteria to help understanding and comparing all the available RL-based routing protocols. These criteria are divided into three groups:

- **context of use**: these are criteria that describe the targeted applications and their characteristics and requirements
- **design characteristics**: criteria in this group highlight how authors designed their protocols to make them efficient and different from the others
- **performance**: in this last category, criteria provide a qualitative evaluation of the overhead of protocols and the metrics used by the authors to analyze simulations

Bibliography

[BL93] Justin Boyan and Michael Littman. "Packet routing in dynamically changing networks: A reinforcement learning approach". In: *Advances in neural information processing systems* 6 (1993).

[NSF+98] Ann Nowe, Kris Steenhaut, Mohamed Fakir, et al. "Q-learning for adaptive load based routing". In: *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218)*. Vol. 4. IEEE. 1998, pp. 3965–3970.