Reinforcement Learning Based Routing in Networks: Review and Classification of Approaches

Alessio Bandiera 1985878

Q-routing

In 1993 Boyan and Littman [BL93] proposed a hop-by-hop routing algorithm based on Q-learning, called **Q-routing**.

Q-routing

In 1993 Boyan and Littman [BL93] proposed a hop-by-hop routing algorithm based on Q-learning, called **Q-routing**.

Most of the existing RL-based routing protocols today are extensions of their work.

Q-routing

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

Q-routing

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

- 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

Q-routing

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

Upon sending packet P to node j^*, node i receives back from node j^* the estimate

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

Q-routing

```
function QROUTING( )
    Initialize Q_i matrix randomly
    while termination condition holds do
        if packet P is ready to be sent to d then
            Determine node j* ← arg min_{j∈N(i)} Q_i(d, j)
            Send packet to node j*
            Collect estimate θ_{j*}(d) from node j*
            Update Q_i(d, j*) ← (1 − α) · Q_i(d, j*) + α · [W_i^q(P) + T_{ij*} + θ_{j*}(d)]
        end if
    end while
end function
```

Then, node i updates $Q_i(d, j^*)$ based on the *update formula* for Q-learning:

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

Q-learning

Despite the wide adoption, Q-routing has some flaws. Some problems are direct consequences of Q-learning such as

- *slow convergence*
- *high parameter setting sensitivity*

However, there are also problems arising from the algorithm itself.

Q-learning

Despite the wide adoption, Q-routing has some flaws. Some problems are direct consequences of Q-learning such as

- *slow convergence*
- *high parameter setting sensitivity*

However, there are also problems arising from the algorithm itself.

For instance the **Q-value freshness**: $\theta_j(d)$ is evaluated only upon packet transmission on a route, therefore if a route is not used for a long time its estimate becomes *outdated*.

Classification criteria

To their knowledge, the authors statae that their work is the first in the literature that proposed **classification criteria** to help comparing all available RL-based routing protocols in the literature.

Classification criteria

To their knowledge, the authors statae that their work is the first in the literature that proposed **classification criteria** to help comparing all available RL-based routing protocols in the literature.

These criteria are divided into 3 groups:

1. **Context of use**: criteria based on the *target applications*
2. **Design characteristics**: criteria based on the *design* of the protocols
3. **Performance**: criteria based on qualitative evaluation on *overhead* and *metrics*

Context of use — Network class and assumptions

TODO

Context of use — Routing optimization context

A *good* protocol should be able to determine and select the optimal paths to convey data from sources to destinations. This can be TODO

Context of use – Unicast or Multicast

Categorizing between **unicast or multicast** approaches is a natural choice, given the inherent *overhead* that multicast routing protocols require.

Context of use – Unicast or Multicast

Categorizing between **unicast or multicast** approaches is a natural choice, given the inherent *overhead* that multicast routing protocols require.

Indeed, RL should be applied in multicasting scenarios only when links are sufficiently stable and/or partial delivery is allowed, otherwise convergence may be outright *impossible*.

Context of use — QoS metrics for optimization

The choice of the metrics is one of the most important aspects of a protocol. When multiple metrics are utilized, they are *weighted* based on the importance — which depends on the target application.

Context of use — QoS metrics for optimization

The choice of the metrics is one of the most important aspects of a protocol. When multiple metrics are utilized, they are *weighted* based on the importance — which depends on the target application.

Quality of Service (QoS) metrics that have been addressed as objectives for RL-based routing include:

Context of use — QoS metrics for optimization

The choice of the metrics is one of the most important aspects of a protocol. When multiple metrics are utilized, they are *weighted* based on the importance — which depends on the target application.

Quality of Service (QoS) metrics that have been addressed as objectives for RL-based routing include:

- **delivery rate**: average time to deliver a packet
- **delivery ratio**: proportion of packets successfully delivered
- **hop count**: average number of hops from source to destination
- **loss ratio**: proportion of packets not delivered

Context of use — QoS metrics for optimization

The choice of the metrics is one of the most important aspects of a protocol. When multiple metrics are utilized, they are *weighted* based on the importance — which depends on the target application.

Quality of Service (QoS) metrics that have been addressed as objectives for RL-based routing include:

- **bandwidth**: average bandwidth provided to sources
- **throughput**: average amount of bytes delivered in the entire network per time unit
- **path stability**: it indicates how a path between source and destination changes over time
- **energy consumption**: average energy consumption of the network

Context of use — QoS metrics for optimization

The choice of the metrics is one of the most important aspects of a protocol. When multiple metrics are utilized, they are *weighted* based on the importance — which depends on the target application.

Quality of Service (QoS) metrics that have been addressed as objectives for RL-based routing include:

- **network lifetime**: average time over which the network is still alive
- **transmission power**: power for performing a transmission
- **hit delay**: average delay to return requrested data in peer-to-peer networks
- **hit ratio**: proportion of satisfied requrests in peer-to-peer networks

Context of use — QoS metrics for optimization

The choice of the metrics is one of the most important aspects of a protocol. When multiple metrics are utilized, they are *weighted* based on the importance — which depends on the target application.

Quality of Service (QoS) metrics that have been addressed as objectives for RL-based routing include:

- **gain**: average revenue (in $) received by the agent — in business contexts
- **overhead**: average *cost* to deliver data packets at destination — the *cost* definition depends on the application

Context of use – QoS guaranteeing

Lastly, a few routing protocols are aimed at providing QoS guarantees, regarding delivery delay to meet some requirements of **delay-sensitive applications**.

Context of use – QoS guaranteeing

Lastly, a few routing protocols are aimed at providing QoS guarantees, regarding delivery delay to meet some requirements of **delay-sensitive applications**.

For instance, this is essential in *multimedia applications*.