

Q-Learning Based Forwarding Strategy in Named Data Networks

Hend Hnaien[1,2(✉)] and Haifa Touati[1,2]

[1] Hatem Bettaher IResCoMath Research Unit, Gabes, Tunisia
haifa.touati@cristal.rnu.tn
[2] Faculty of Science of Gabes, Gabes, Tunisia
h2II_05@hotmail.com

Abstract. Named Data Networking (NDN) emerged as a promising new communication architecture aimed to cope with the need for efficient and robust data dissemination. NDN forwarding strategy plays a significant role for efficient data dissemination. Most of the currently deployed forwarding strategies use fixed control rules given by the routing layer. Obviously these simplified rules are inaccurate in dynamically changing networks. In this paper, we propose a novel Interest forwarding scheme called Q-Learning based Forwarding Strategy (QLFS). QLFS embedded a continual and online learning process that ensures quick reaction to sudden disruption during network operation. At each NDN router, forwarding decisions are continually adapted according to delivery times variation and perceived events, i. e. NACK reception, Interest Timeout... Our simulation results show that our proposed approach is more efficient than state of the art forwarding strategy in term of data delivery and number of timeout events.

Keywords: Named data networking · Forwarding strategy · Q-learning

1 Introduction

Named Data Networking [1] is a new data-centric architecture that redesigns the Internet communication paradigm from IP address based model to content name based model. The main idea of the Named Data Networks communication model is to focus on the data itself instead of its location. In Named Data Networks, users insert the name of the requested content into a request packet, called *"Interest"*, that is sent over the network and the nearest node which has previously stored this content, or *"Data"* packet, must deliver it. In NDN, the node that sends the *Interest*, is called *Consumer* and the original *Data* source is called *Producer* [2].

Each NDN node manages three different data structures: The *Pending Interest Table (PIT)*, the *Content Store (CS)* and the *Forwarding Information Base (FIB)*. The *Pending Interest Table (PIT)* tracks the outstanding Interests as

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12249, pp. 434–444, 2020.
https://doi.org/10.1007/978-3-030-58799-4_32

well as the interface from which they come, in order to deliver back the Data to the Consumer on the Interest reverse path. The *Content Store (CS)* is used to store the received Data packets in order to serve the upcoming demands of the same Data. Finally, the *Forwarding Information Base (FIB)* stores the Data name prefixes and the interfaces where to forward the *Interest* packets. The FIB is populated and updated by a name-based routing protocol. For each incoming Interest, a longest prefix match lookup is performed on the FIB, and the list of outgoing faces stored on the found FIB entry is an important reference for the strategy module.

To request a content, the consumer sends an Interest and based on the forwarding strategy, the Interest packet is passed from one node to another until it reaches the producer that holds the requested content. The simplest forwarding strategy is the *multicast approach* [3] that forwards incoming Interest to all upstreams indicated by the supplied FIB entry, except the incoming interface. The *best route* [3] strategy forwards interests using the eligible upstream with the lowest routing costs. If there is no available upstream, e.g. the upstream link is down, the Interest is rejected and a *NACK* with the error code *no route* is generated and returned to the downstream/consumer node via the incoming interface of the Interest. If the consumer wants, it can retransmit the Interest, and the *best route* strategy would retry it with another next hop.

However, basic forwarding strategies do not adapt to the dynamic nature of the network. They frequently cause problems when there is a sudden change in the network architecture. Several forwarding strategies have been proposed in the literature to adapt the forwarding strategy to different network environments [4], [5] and [6]. In this context, we have thought to exploit reinforcement learning, which has proven to be efficient for real time decision making in different sectors and even for network traffic management.

The main contribution of this paper is the proposal of a new adaptive interest forwarding strategy for Named Data Networks. This strategy is based on the notion of reinforcement learning [7]. During execution, each node goes through training and exploitation phases. In the training phase, each node tries to have a background on its neighborhood and on the best path, in term of delivery time and reliability, to reach a given content. Once the environment has been explored, the node transits to the exploitation phase, and interest forwarding is done based on the $Q - values$ already calculated in the previous phase.

The remaining of the paper is organized as follows: In Sect. 2, we give a brief presentation of the Q-Learning process. Then, we present our solution called QLFS in Sect. 3. In Sect. 4, we move on to evaluate our solution by comparing its performances to the *best route* strategy. Finally, Sect. 5 concludes the paper and gives some future directions of this work.

2 An Overview of the Q-Learning Technique

Q-Learning [8] is a reinforcement learning algorithm that seeks to find the best action to take given the current state. The Q-Learning process involves 5 key

entities: an **Environment**, an **Agent**, a set of **States** S, **Reward** values, and a set of **Actions** per state, denoted A. By performing an *Action* $a_{i,j} \in A$, the *Agent* transits from a *State* i to a *State* j. Executing an *Action* in a specific *State* provides the *Agent* with a *Reward*.

The agent is immersed in an environment, and makes decisions based on his current state. In return, the environment provides the agent with a reward, which can be positive or negative. The agent seeks, through iterated experiences, an optimal decision that maximizes the sum of the rewards over time.

It does this by adding the maximum reward attainable from future states to the reward achieved by its current state. Hence, the current action is influenced by the potential future reward. This potential reward is a weighted sum of the expected rewards values of all future steps starting from the current state.

3 Q-Learning Based Forwarding Strategy (QLFS)

The main idea of our solution is to develop a Q-Learning based forwarding strategy that helps the NDN router to find the suitable next hop and to adapt quickly its forwarding decisions according to network conditions.

As explained in the previous section, to apply Q-Learning we need to define five principals components which are: Environment, Agent, States, Actions and Rewards. In the context of Interest forwarding, we choose to model these components as follows:

- The *Environment* is the Named Data Network.
- Each NDN node is an *Agent*.
- A *State* is the next hop to which a received interest is forwarded.
- An *Action* is forwarding an interest to one of the neighboring nodes.
- *Rewards* are RTT based values.

To decide to which adjacent node should an NDN router forwards the received interest to reach as quickly as possible the content producer, a new $Q-table$ structure is implemented and maintained at each node. This table holds the $Q-value$ corresponding to each decision. For each forwarded interest, the router waits for the corresponding data packet to update the $Q-values$. On the arrival of a Data packet, a reward value is computed for the arrival interface and used to update the $Q-value$ as follows:

$$Q_n(x,c) = (1-\alpha) * Q_{n-1}(x,c) + \alpha * [R_n(x,c) + \max_{v \in N(x)} Q_n(v,c)] \quad (1)$$

where:

- $Q_n(x,c)$ is the $Q-value$ to reach a content c from a node x during the n^{th} iteration.

- $R_n(x, c)$ is the reward value for the transition from node x to the next node selected to reach the content c. This value depends on the delivery time and the loss and failure events. The router computes the delay between the transmission of an interest and the reception of the corresponding data packet, denoted RTT. When the data packet is received, the current RTT value is compared to the minimum, average and maximum observed RTT values and the reward value is set according to the interval where the current RTT is. The closer the current RTT is to the minimum RTT, the higher the reward will be and vice versa: the closer the current RTT is to the maximum RTT, the lower the reward will be. Moreover, if a timeout event is detected or a NACK packet is received a penalty is applied by assigning a negative reward value.
- $Q_n(v, c)$ is the $Q - value$ to reach a content c from a neighbor node v during the n^{th} iteration.
- $N(x)$ is the list of neighbors of node x.
- α is the learning rate which controls how quickly the algorithm adapts to the random changes imposed by the environment.

We note, that we modified the Data packet structure, as illustrated in Fig. 1, by adding a new field to hold the maximum $Q - value$ for a given prefix and return it to the downstream neighbor. In fact, before returning a data packet, the router node looks for the next hop that has the highest $Q - value$ and inserts this value in the data packet. For this purpose we have defined a new tag, called *MaxQTag*, so that we can save the maximum $Q - value$ among all the $Q - values$ of neighboring nodes.

When a node receives a data packet, it extracts the tag and uses it to update the $Q - value$ relative to this prefix, according to Eq. 1.



Fig. 1. Modified data packet structure

When an Interest is received, the router forwards it according to the current phase. Three phases are defined in QLFS: the **Initial training phase**, the **Training phase** and the **Exploitation phase**.

- During the **Initial training phase**, each received interest will be:
  - Forwarded to the lowest cost next hop stored in the FIB table. This action assures that the interest reaches the content producer, which reduces the loss probability.
  - Forwarded to one of the unused next hops. There is no random choice of next hop thus we can explore more possible links.

– During the **Training phase**, the router performs two actions:
  - Instead of using the routing cost metric to make decision, the node chooses the face with the maximum $Q - value$ to forward the interest.
  - The node continues to explore other paths by forwarding the interest to one of the unused next hops.

  Before leaving the training phase, the maximum $Q - value$, $Q_{max}(x, c)$, is saved for each node.
– During the **Exploitation phase**, each router forwards the received interest to the next hop that has the maximum $Q - value$. In fact at this step: each node has the necessary knowledge-background about how to reach a given content and which link to choose based on $Q - values$.

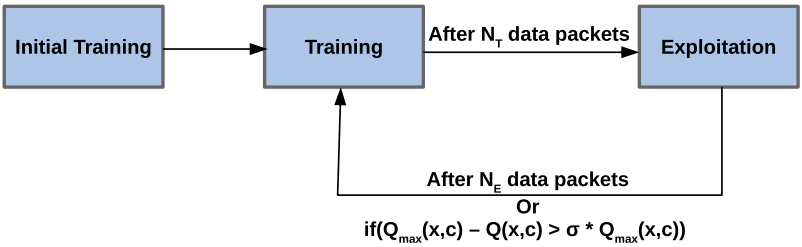


Fig. 2. QLFS learning cycle

An NDN router switches continually between the *training* and the *exploitation* phases. As shown in Fig. 2, after the reception of N_T data packets, the algorithm switches from *training* to *exploitation*. During the exploitation phase, only the maximum $Q - values$ will be updated. To detect eventual change in network conditions (failed link repaired, other new added links to explore...), our system periodically returns to the *training* phase. More precisely, QLFS switches to the *training* step, if N_E data packets are received or the performance of the chosen best link degrades. This latter is detected through the following test:

$$Q_{max}(x, c) - Q(x, c) > \sigma * Q_{max}(x, c) \tag{2}$$

In summary, during the *exploitation* phase, for each update of the $Q - table$, the QLFS strategy verifies the number of data packets received during the current *exploitation* phase and the gap between the current $Q - value$ and the $Q_{max}(x, c)$ used at the beginning of the current *exploitation* phase. If one of the above tests is true, the QLFS protocol returns to the *training* phase.

4 Performance Evaluation

In this section, we evaluate the proposed QLFS strategy by comparing it with the basic forwarding strategy of the NDN architecture, namely *best route*. To that

end, we have developed our proposal on ndnSIM [9], which is an NS-3 based simulator that implements the Named Data Networking (NDN) architecture using C++ language.

4.1 Simulation Model

Our simulation model consists of five nodes: one consumer, one producer and three routers as shown in Fig. 3. The nodes are connected through Point-To-Point links. Each link is characterized by a data rate of 1Mbps, a delay of 10 ms and a queue size of 20 packets.

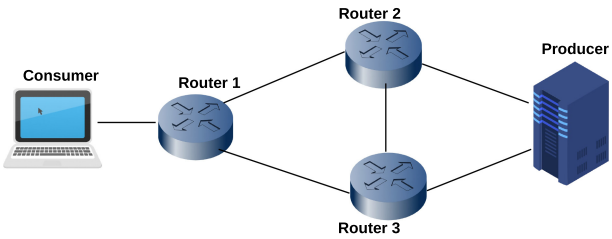


Fig. 3. Simulation model

The consumer sends 100 interests per second and the producer will reply to all requests with data packets of size 1024 KB. Table 1 lists most of the simulation parameters.

Table 1. Simulation Parameters.

Parameter	Value
Network links	P2P links
Data Rate	1 Mbps
Delay	10 ms
Payload Size	1024 Kb
Queue	20 packets
Interest Rate	100 interests per second
Simulation Time	20s
α	0.8
σ	0.5

In order to better set the value of the parameter *learning rate*, denoted α, we have run simulations while varying the value of this parameter from 0.2 to 0.9. Results reported in Fig. 4, show that the best performances are achieved when

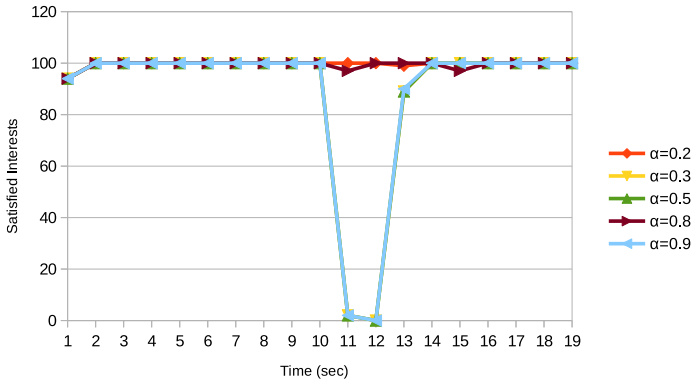


Fig. 4. Setting the learning rate parameter

the *learning rate* is set to 0.8. Thus this value will be used in the subsequent simulations.

To study the performance of the proposed QLFS scheme, we used the following metrics: *Satisfied interests*, *TimedOut Interests* and *Out Interests per interface*. The evaluation metrics are defined as follows:

- **Satisfied interests**: This metric is computed at the consumer side and is defined as the number of interest packets for which the consumer receives the requested data packets.
- **TimedOut Interests**: This metric is computed at the consumer side and is defined as the number of interests for which the consumer hasn't received the requested data until lifetime expiry.
- **Out Interests per interface**: This metric is computed by router *Router*1 and is defined as the number of interest forwarded through a each interface of the router.

4.2 Simulation Results and Performance Analysis

In a first part, we will compare the two strategies, QLFS and *best route*, following the metrics already mentioned in an ideal scenario, i.e. without link failure. In the second part, we compare the performance of both strategies in a dynamically changing network, i.e. in presence of link failure.

Scenario Without Link Failure. Simulations performed is a scenario without link failure, show that the two strategies have similar performances in terms of *satisfied interests* (Fig. 5) and *timedout interests* (Fig. 6), these results confirm that our solution doesn't degrade the network performance when link quality is stable and no failure or disruptive event occurs.

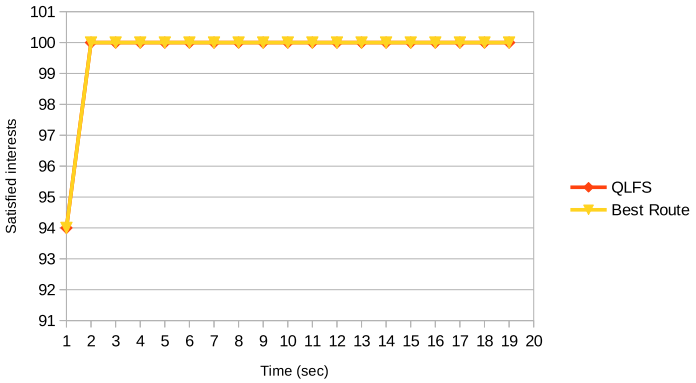


Fig. 5. Satisfied Interest in a scenario without link failure

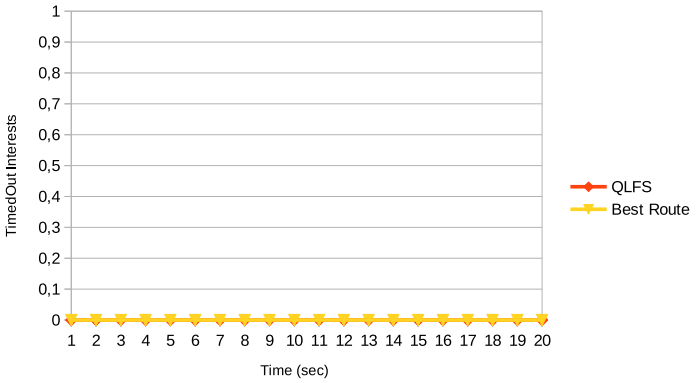


Fig. 6. TimedOut Interest in a scenario without link failure

Scenario with Link Failure. In a real-world scenario, one or more failed links are probably encountered. For this purpose, we provoke a link failure between *Router*1 and *Router*3 and study how each approach will react to this sudden and temporary disruption.

The evaluation of the number of Out Interests per interface at *Router*1, reported in Fig. 7, show that using the *est route* strategy, *Router*1 forwards all interests through interface 258 which is linked to *Router*3. Even when the destination becomes inaccessible from this interface after the link failure, *best route* continues to choose this interface despite the presence of a better alternative through interface 259, which is linked to *Router*2.

However, using the QLFS strategy, and as shown in Fig. 8, when the link failure occurs, *Router*1 immediately switches and forwards all interests through interface 259. The adaptive learning process introduced in QLFS helps the *Router*1 to quickly detect the failure event, through the penalty applied to interface 258. This penalty reduces the $Q - value$ of interface 258 and favours sending the interests through another interface, namely interface 259 which has better $Q - value$.

These results, confirm that the QLFS strategy is more efficient and more reactive to sudden disruptions than the *best route* strategy.

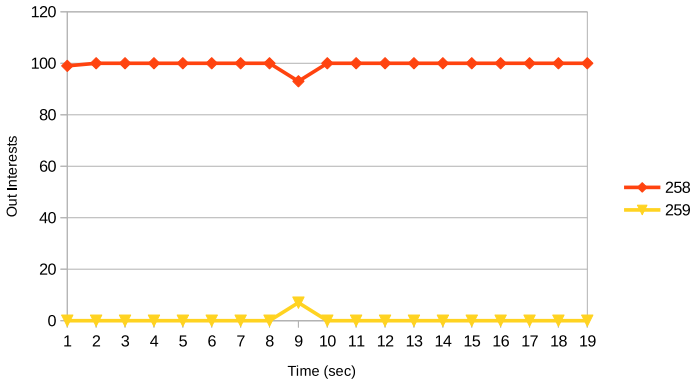


Fig. 7. Best route: out interests from router 1 in a scenario with link failure

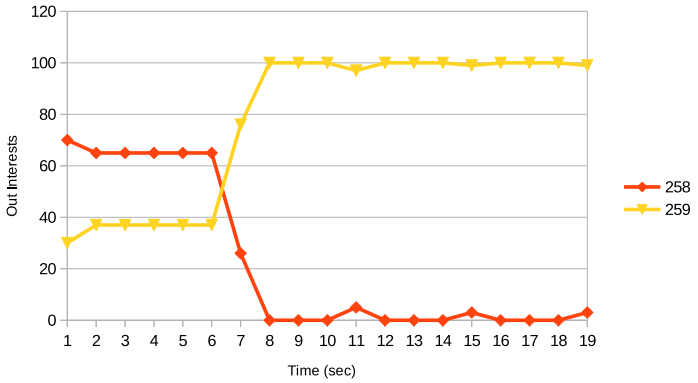


Fig. 8. QLFS: out interests from router 1 in a scenario with link failure

Finally, the evaluation of the number of satisfied interest reported in Fig. 9, reveals that when the link failure is applied, the number of satisfied interest decreases to 0 using *best route*, but using QLFS the number of satisfied interest is not affected by the presence of link failure. Similarly, Fig. 10 shows clearly that using *best route* several interest timeout events occur which induce unnecessary interest retransmissions and waste network bandwidth. However, using QLFS, the number of timed out interests tends towards 0 and thus the network bandwidth is efficiently consumed.

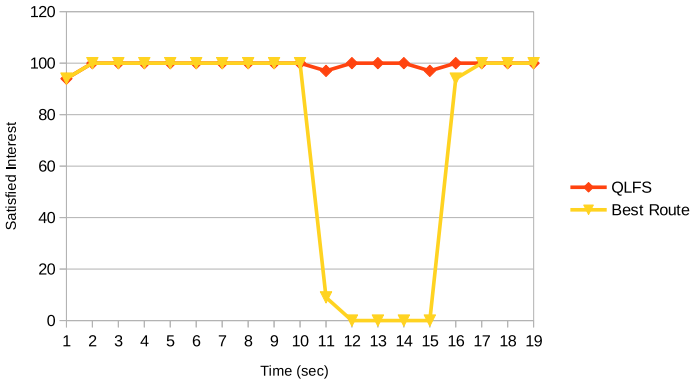


Fig. 9. Satisfied Interests in a scenario with link failure

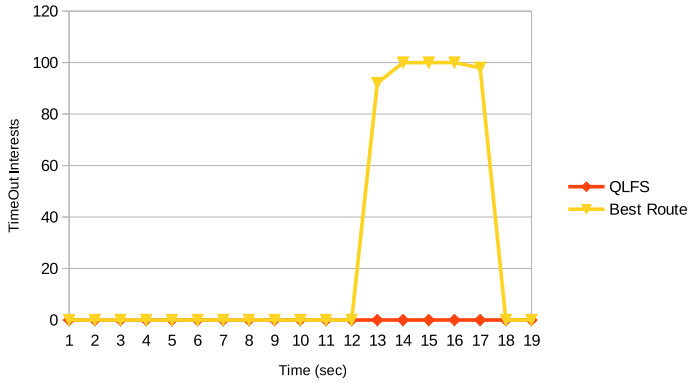


Fig. 10. TimedOut Interests in a scenario with link failure

As a conclusion, all simulations results confirm that the performance of the *best route* strategy degrades as the link already chosen by this strategy is broken, whereas the QLFS strategy is more resistant to this problem and can easily and rapidly find an alternative path to forward interest packets.

5 Conclusion

In this paper, we proposed the QLFS strategy, which brings more efficiency and reactivity to the interest forwarding process in the NDN architecture. Through the use of reinforcement learning and more precisely the Q-Learning algorithm, QLFS performs an online learning to adapt its forwarding decision according to the delivery time and the perceived loss events. Analysis and simulations were conducted to evaluate the performance of the proposed strategy. The results show that QLFS outperforms *best route* in term of speed of finding an alternative path when the initial path is interrupted. Results, also show that QLFS significantly

minimizes the number of timed out interests, hence maximizing the number of satisfied interests compared to the *best route* strategy. In a future work, we plan to extend our experimental study to more complex scenarios.

References

1. Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N.H., Braynard, R.L.: Networking named content. In: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, New York, NY, USA, pp. 1–12 (2009)
2. Wissingh, B., Wood, C., Afanasyev, A., Zhang, L., Oran, D., Tschudin, C.: Information-Centric Networking (ICN): Content-Centric Networking (CCNx) and Named Data Networking (NDN) Terminology. RFC 8793, June 2020
3. Yi, C., Afanasyev, A., Moiseenko, I., Wang, L., Zhang, B., Zhang, L.: A case for stateful forwarding plane. Comput. Commun. **36**, 779–791 (2013)
4. Kardi, A., Touati, H.: NDVN: named data for vehicular networking. Int. J. Eng. Res. Technol. IJERT **4**, 1–6 (2015)
5. Aboud, A., Touati, H.: Geographic interest forwarding in NDN-based wireless sensor networks. In: Proceedings of the 13th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA), Morocco, Agadir, pp. 1–8 (2016)
6. Aboud, A., Touati, H., Hnich, B.: Efficient forwarding strategy in a NDN-based internet of things. Cluster Comput. **22**(3), 805–818 (2018). https://doi.org/10.1007/s10586-018-2859-7
7. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: a survey. J. Artif. Intell. Res. **4**, 237–285 (1996)
8. Watkins, C.J., Dayan, P.: Technical note: Q-learning. Mach. Learn. **8**, 279–292 (1992)
9. Mastorakis, S., Afanasyev, A., Moiseenko, I., Zhang, L.: ndnSIM 2: an updated NDN simulator for NS-3. Technical report NDN-0028, Revision 2 (2016)