AC-QoS-FS: Ant colony based QoS-aware forwarding strategy for routing in Named Data Networking

Abdelali Kerrouche
Image, Signal and Intelligent Systems
Laboratory, University of Paris-Est, France
Email: abdelali.kerrouche@univ-paris-est.fr

Mustapha Reda Senouci
Ecole Militaire Polytechnique
Algiers, Algeria
Email: mrsenouci@gmail.com

Abdelhamid Mellouk
Image, Signal and Intelligent Systems
Laboratory, University of Paris-Est, France
Email: mellouk@u-pec.fr

Thiago Abreu
Image, Signal and Intelligent Systems
Laboratory, University of Paris-Est, France
Email: thiago.abreu@irisa.fr

***Abstract*—This paper proposes a new QoS-aware forwarding strategy for Named Data Networking. Borrowing techniques from the ant colony optimization, the proposed strategy, which is called Ant colony based QoS-aware forwarding strategy (AC-QoS-FS), makes full use of both forward and backward ants to rank interfaces. Forward and backward ants (Interest and Data packets) probe real-time network QoS parameters to update the interfaces ranking in order to select the best one for forwarding the incoming Interests. The effectiveness of AC-QoS-FS is validated through ndnSIM simulation.**

Keywords: Information-Centric Networking, Named-Data Networking, Routing, Forwarding, Ant colony optimization.

I. Introduction

Nowadays, the Internet is experiencing a shift from host-centric communications paradigm to a data-centric communications one. In the former case, the location of a content was more significant than the content itself, which the latter case intends to change. Among the earlier attempts to perform such modification, notably the Peer-to-Peer (P2P) networks and Content Delivery Networks (CDN) shall be mentioned, in which the necessary change for the shift of paradigm was performed at the application layer. Many studies show that P2P and CDN can enhance QoS (Quality-of-Service) and QoE (Quality-of-Experience) of end-users. However, they incur high costs and may lead to inefficient solutions. Therefore, more recent attempts such as NDN (Named-Data Networking) architecture focus on network layer approaches [1].

The communications in NDN are based on two types of packets: Interest and Data. The former is a small packet identified by a given name, rather than IP addresses, containing a request generated by the user, while the latter corresponds to the desired data. The consumer sends an Interest packet with the name of the content through the network and waits that the producer of the content replies with a Data packet containing the requested data along with some additional information such as the signature. Nodes in the path from producer toward consumer may store a Data packet (in a repository or in its cache), in order to send it whenever other Interests for that same content arrive.

The routing mechanism in NDN takes advantage of the data names, which are structured hierarchically, in order to simplify the transmission of data [2]. Data satisfies an Interest if the data name in the Interest packet is the same or is a prefix of the data name in the data packet.

Three structures compose each NDN node: **CS** (Content Store), which stores temporarily the data received at the node, preventing useless forwarding of Interests; **PIT** (Pending Interest Table) which keeps track of pending Interests that were forwarded and not yet satisfied; and **FIB** (Forwarding Information Base), a table for data name prefixes used to forward Interest packets toward potential content producers. Note that in NDN networks, the routers can forward received packets toward multiple interfaces, according to the entries in the FIB table.

How to support smart forwarding of Interests over multipaths while considering QoS parameters is a challenging issue in NDN. To address this challenge, this paper proposes Ant colony based QoS-aware forwarding strategy (AC-QoS-FS). In comparison to related work, AC-QoS-FS enhances data delivery performance and traffic distribution while avoiding network instability.

The rest of this paper is organized as follows. The proposed QoS-aware forwarding strategy AC-QoS-FS is described in Section II and its performances are evaluated in Section III. Section IV reviews related work. Finally, Section V concludes the paper and discusses future work.

II. Our Proposal: the Ant Colony Based QoS-aware Forwarding Strategy

We point out that the NDN architectures present a symmetric packet routing, i.e., Interest packets and the respective Data packets flow through the same path, but in opposite directions. This behavior matches the natural behavior of ants while searching for shortest path between their nest and food source. Based on this idea, the ant colony algorithm (the ant system) was originally used to solve the famous NP-complete Traveling Salesman Problem (TSP). Following the same principle, we adapted this

978-1-4673-8999-0/17/$31.00 ©2017 IEEE

algorithm to the design of our proposed forwarding strategy to solve the issues of QoS-aware routing in NDN networks.

The main idea of our proposal relies on ants measuring the real-time network QoS parameters of the path they traversed from the data-source (producer) up to the data requester (consumer), and using these measurements to compute the amount of pheromone to be deposited in order to select the best interface to forward the incoming Interests. When arriving at each node of the path, an ant deposits a certain amount of pheromone for that correspondent data incoming interface, according to the measured QoS parameters of the path. These amounts of pheromone will be used for ranking interfaces in order to select the best one for forwarding the incoming Interests. The ultimate goals that we aspire by the proposed forwarding strategy are: i) to maintain a best data delivery performance, ii) to get a best traffic distribution over the network links, and iii) to avoid network instability. Our proposed Ant Colony based QoS-aware Forwarding Strategy for NDN will be called **AC-QoS-FS** and it is detailed hereafter.

A. Designing AC-QoS-FS

Let $G(X,U)$ be the graph representing the network topology where X ($|X| > 0$) is a set of NDN nodes, and U ($|U| > 0$) is a set of links connecting such nodes. The search space is the set of the available paths in the graph $G(X,U)$ representing the NDN network. There are three kinds of nodes:

- **Consumer (nest)**: it is similar to an ant nest. In our proposal there are several nests, where they express their food needs that match the data by their names.
- **Producer (food source)**: a data storage server able to respond to various arriving demands. It represents the foods source, where ants can get their food needs (data).
- **Router (city)**: it is responsible to direct the Interests packets towards the potential sources of the requested data at first hand, i.e. the Interest ants to the food source, and directs the Data packets toward the appropriate consumer, by exploiting data names, i.e. data ants to the nest. It is a seam connecting cities in TSP. It makes difference between packets.

In our proposal, the ant is a packet identified by the data name, being either an Interest packet in the search phase of requested data, or a Data packet, after meeting the requested data, upon the transmission data phase toward the consumer(s). Table I relates the nomenclatures of NDN and Ant Colony and TSP models.

NDN Network	**Ant Colony**
Packet (Interest - Data)	Ant
Consumer	Nest
Producer or caching router	Food source
Routers	Cities
Links	Routes between Cities

TABLE I: Relation between NDN and Ant Colony.

Three QoS parameters are considered: *cost*, *bandwidth* (BW), and *Round Trip Time* (*RTT*). Therefore, according to the values of these parameters, the forwarding decisions for Interests and Data packets will be performed. Real ants have a very limited memory. The virtual ants used for the TSP algorithm, however, can store the cities list that they traversed. In our proposal, the ant has also a memory, which allows it to store the network QoS measurements of the path traversed from the food source toward the nest. In other words, each ant, when returning toward its nest, stores the residual bandwidth and the cost of the path traversed between food source and last node. This information will be used to compute the pheromone amount that will be deposited in the FIB entry corresponding to the data handled by the ant (in the field corresponding to the interface of the incoming data). Hereafter, we detail how these parameters are estimated.

1) Bandwidth: The proposed forwarding strategy manages available links bandwidth from interfaces. Thus, the capacity $C1$ of the link that connects routers X to Y by using interface i, is managed via the interface i and the capacity $C2$ of the link that connects routers Y to X via j is managed via interface j.

$$\left| \ BW(X)_{int-i-Total} \right| = C1$$

$$\left| \ BW(Y)_{int-j-Total} \right| = C2$$

where $BW(X)_{int-i-Total}$ is the bandwidth capacity of the interface i of the router X and $BW(Y)_{int-j-Total}$ is the bandwidth capacity of the interface j of the router Y. We have also:

$$\left| \ BW(X)_{int-i-res} \right| = BW_{link_{res}(X,Y)}$$

$$\left| \ BW(Y)_{int-j-res} \right| = BW_{link_{res}(Y,X)}$$

where $BW(X)_{int-i-res}$ is the residual bandwidth of the interface i of the router X that connects X to Y, and $BW_{link_{res}(X,Y)}$ is the residual bandwidth of the link that connects X via i to Y, where $BW(Y)_{int-j-res}$ is the residual bandwidth of the interface j of the router Y that connects Y to X, and $BW_{link_{res}(Y,X)}$ is the residual bandwidth of the link that connects Y to X via interface j.

The proposed forwarding strategy allocates the necessary bandwidth resource from outgoing interfaces, before sending the incoming Interest to the next hop on upstream. This process will be performed by all routers traversed by the incoming Interest for all links of the path connecting the consumer to the data-source.

1.1) Residual bandwidth of links:
Let us consider the link that connects the node X to node Y via the interface i. The residual bandwidth of this link is equal to the residual bandwidth of the interface i. It can be calculated, when n Interests are sent from X to Y via the interface i in a given time t as follow:

$$\left| \ BW(X)_{int-i-res} \right| = BW_{link_{res}(X,Y)} = \left| \ BW(X)_{int-i-Total} \right| - \sum_{k=1}^{n} \left| \ BW_{req_k} \right| \quad (1)$$

where BW_{Req_k} is the necessary bandwidth for the Interest k which must be reserved.

1.2) The path residual bandwidth:
Let us assume that an ant (Interest) has been forwarded from node R_i to the next node R_{i+1} on upstream via the interface k selected by the forwarding strategy. While the ant (Data) returns back to the node R_i, it also brings information concerning the quality of the path traversed from the R_{i+1} to R_i, such as the minimum bandwidth T of the current link. This information is used to determine the amount of pheromone that the ant will depose at the interface connecting such link.

From Figure 1, the path between a node $R_i, i \in [1, n]$ and the data source (R_n) and the residual bandwidth of an interface k of the node R_i, $i \in [1, n]$ for a data name prefix P in the FIB table, can be calculated as follows:

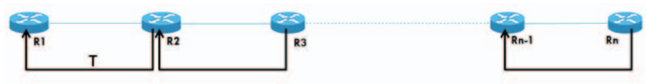


Fig. 1: Updating the residual bandwidth of a path.

$$\left| BW(R_i, P)_{int-k-path} \right| = \min(\left| BW(R_i)_{int-k-res} \right|, T) \quad (2)$$

where:

$$T = \min(\left| BW(R_{i+1})_{int-x-res} \right|, \left| BW(R_{i+2})_{int-k-path} \right|) \quad (3)$$

1.3) The path residual bandwidth update:
In our forwarding strategy, bandwidth updating process is triggered by the events: receiving an Interest or a Data packet.

1.3.1) After receiving an Interest:
The event of reception of an Interest triggers the selection interface process. Once the interface is selected, the forwarding strategy reserves the necessary bandwidth from this interface. Furthermore, it sends the Interest to the upstream node. The reservation is performed as follows:

$$\left| BW(X)_{int-k-res} \right| = \left| BW(X)_{int-k-res} \right| - \left| BW_{req} \right| \quad (4)$$

where k is the selected interface and $\left| BW_{req} \right|$ is the necessary bandwidth.

1.3.2) After receiving a data:
The proposed forwarding strategy releases the reserved bandwidth from the incoming interface of the ant and updates, in the FIB table entry corresponding to the incoming data name (in the field that matches the incoming interface of the ant), the residual bandwidth of the path that connects the current node to the source of the data. The releasing is performed as follows:

$$\left| BW(X)_{int-k-res} \right| = \left| BW(X)_{int-k-res} \right| + \left| BW_{req} \right| \quad (5)$$

where k is the incoming interface of the ant and $\left| BW_{req} \right|$ is the bandwidth reserved at the search phase. The estimated path residual bandwidth, if T is the minimum bandwidth of the links that the ant traversed from the food-source to the current node, is computed as:

$$\left| BW(X)_{int-k-path} \right| = \min(\left| BW(X)_{int-k-res} \right|, T) \quad (6)$$

2) Cost: Let us assume that the cost of the link connecting the nodes X and Y via the interface i equals the cost of the interface i. The cost of the path that was traversed by the ant is calculated by adding up the cost of links in that path. Further, it is also the sum of the nodes incoming interfaces costs that form the path. The cost of the path from the source food to the current node X for k in P is computed as follows:

$$CostP(X, P, k) = T + Cost(X, Y, k) \quad (7)$$

where k is the incoming interface of the ant, and P is the FIB data name prefix entry that matches the incoming data handled by the ant in the node X to the next hop node on upstream Y via k. $Cost(X, Y, k)$ is the cost of the link that connects the node X to the node Y via the interface k and T is the cost of the path that connects Y to the source food.

3) Round Trip Time (RTT): For a given prefix P in the FIB table of a node X, RTT is the time elapsed between the Interest sending and Data reception via the interface k. In this proposal, this is computed by a timer starting when sending the ant to the next node on upstream and stopping after receiving the ant that handled the requested data from the same interface k.

$$RTT(X, P, k) = Time_{Receiving} - Time_{Forwarding} \quad (8)$$

To implement the proposed strategy, it is necessary to adapt the node data structures to our forwarding strategy design requirements. Consequently, we add other fields to the node data structures: one field is used to store the pheromone and the other to save the network QoS parameters metrics collected by the ants. The added fields are (see Figure 2):

- **Pheromone**: it stores the amount of pheromone;
- **RTT**: it stores the round trip time elapsed between the departure of the ant and its return from the same interface;
- **BW** and **Cost** are used, respectively, to store the residual bandwidth and cost of the path traversed by the ant from the food source to the current node.

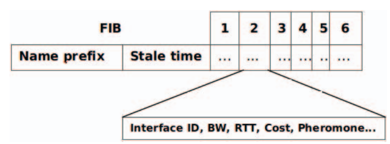


Fig. 2: Forwarding state in FIB.

4) The moving rule: An Interest (ant) located in the node x_i can be forwarded toward one among all nodes connected to x_i. The AC-QoS-FS uses the pheromone amount $\tau_{i,j}$ of the interface that connected x_i to x_j to compute the probability of the node x_j as next hop. Initially, the pheromone amount is $\tau_{i,j} = \tau_0$. If N_{x_i} is a set of nodes connected to x_i except for the node where the incoming Interest comes from ($|N_{x_i}| \geqq 1$), the probability of sending the incoming Interest via the interface j connected to the node x_j is mathematically written as follows:

$$P_{i,j} = \begin{cases} \frac{\tau_{i,j}}{\Sigma_{j \in N_{x_i}} \tau_{i,j}}, & \text{if } j \in N_{x_i} \\ 0, & \text{if} j \notin N_{x_i} \end{cases} \quad (9)$$

where the transition probabilities $P_{i,j}$ of the node x_i fulfill the constraint $\Sigma_{j \in N_{x_i}} P_{i,j} = 1$, $i \in [1, |X|]$.

5) Pheromone update: The real ant deposits the pheromone on all the path when it returns back to its nest. In ant colony based TSP algorithm, the ants update the pheromone amount when all the ants finished a cycle (when all the cities have been visited by the ants). In our proposal, when the ant finds (meets) the requested data, it goes back to the consumer node (starting point or nest), taking the same path traversed during the research phase in opposite direction. Consequently, it crosses the same nodes but in reverse order. Therefore, at each node, it updates the amount of pheromone according to the quality of the path traversed from the food source (based on the QoS parameters), which can be a producer or a caching router (content store), to the current node, only in the field corresponding to the incoming interface of the ant in the FIB entry matching the data name handled by the ant.

Consider that a data (ant) arriving at node x_i through interface K matches the prefix P in the FIB table. The amount of pheromone that the ant will deposit (update) on the correspondent field is computed as follows:

$$\tau_{x_i}(P,k)(t+1) = (1-\rho)\tau_{x_i}(P,K)(t) + \rho\Delta\tau_{x_i}(P,K) \quad (10)$$

$\tau_{x_i}(P,k)$ is the pheromone amount of the interface k for the prefix P in the node x_i. $\Delta\tau_{x_i}(P,k)$ is the amount of pheromone that the ant will deposit on the interface k of the node x_i for the prefix P.

$$\Delta\tau_{x_i}(P,k) = \frac{Q}{\alpha * RTT + \dfrac{\beta}{BW} + \gamma * cost} \quad (11)$$

with $\alpha + \beta + \gamma = 1$, α, β and γ are the weighting constants that represent respectively the importance of the RTT, bandwidth and cost. Q is a tuning parameter that will be discussed in Section III. The amount of pheromone that the ant will deposit cannot be computed by equation (11) because it groups two kind of parameters: BW, which is beneficial criterion whereas the others parameters RTT and cost are expense criteria. So, the values of parameters need to be normalized. The expense criteria are normalized as $r_{i,j} = \frac{V_{min}}{V_{i,j}}$. Where V_{min} is the smallest value of the criterion j for all the interfaces and $V_{i,j}$ is the values of the criterion j for the interface i. Furthermore, RTT and cost are normalized respectively as $\frac{BRTT}{RTT}$ and $\frac{BCost}{Cost}$. $BRTT$ is the best (min) RTT of all the interfaces and RTT is the RTT of the interface. $BCost$ is the best (min) cost of all the interfaces and $Cost$ is the cost of the incoming interface of ant. Equation (11) will be redefined as follows:

$$\Delta\tau_{x_i}(P,k) = \frac{Q}{\alpha * \frac{BRTT}{RTT} + \beta * \frac{BW}{BBW} + \gamma * \frac{BCost}{Cost}} \quad (12)$$

B. Forwarding decision-making algorithms

In this section, we detail how AC-QoS-FS reacts to the main events: reception of Interest and Data. Furthermore, we present the algorithms that follow upon reception of those packets.

1) After receiving an Interest: The proposed forwarding strategy performs the following steps. First, it checks the Content Store (CS) for data matching the incoming Interest name. If matched data is found, then it will be sent on downstream via the incoming interface of the Interest and the Interest will be dropped as it is considered satisfied. If no data were found in the CS, the node looks in PIT entries. If an entry is found, the forwarding strategy updates the bandwidth and adds the incoming interface to the requesting interfaces list, then it drops the Interest. When the requested data is found, a copy will be sent back via incoming interfaces of Interest. If the Interest name does not exist in the PIT, our forwarding strategy creates a new PIT entry and looks up for the Interest name in the FIB table. If no FIB entry matches the Interest name, the Interest is deleted. Otherwise, there is a FIB entry that matches the requested data name, AC-QoS-FS will forward the incoming Interest as explained in II-A4.

In order to get the best data delivery performance and avoid network instability, the proposed forwarding strategy does not forward the incoming Interest via the interface that has the biggest probability. Instead it uses the probabilities computed for all interfaces as inputs for *the roulette wheel selection algorithm*, which makes an oriented randomly selection corresponding to the interface probability. Consequently, the proposed forwarding strategy does not always send the incoming Interests via the interface that has the biggest probability (the biggest pheromone amount); however, it gives opportunity to other interfaces that have a small probability (pheromone), which ensures better traffic distribution over the network links.

Once the selection process is finished and the forwarding interface is selected, the proposed forwarding strategy starts by reserving the necessary bandwidth from the selected interface, then it updates the PIT entry with the selected interface as outgoing interface and finally it sends the Interest. Algorithm 1 explains in detail through a pseudo-code how our forwarding strategy works when a router receives a new incoming Interest.

```
if ((Interest.name) is found in CS) then
    Forward(ant, k)
else
    if ((Interest.name) is found in PIT) then
        add(Requesting Faces list, k)
    else
        if ((Interest.name) is found in FIB) then
            Create new PIT entry (Interest.name, nonce, interface, k)
            *** Selection process ***
            for (each face K ∈ FIB Entry) do
                Compute the probability of interface K according to
                  Equation 9
            end
            Use the roulette wheel to select the out interface S
            *** Forwarding process ***
            | BW_int-S-res | ← | BW_int-S-res | − | BW_req |
            Update PIT entry with S as out interface
            Send(Interest, S)
        else
            Drop Interest
        end
    end
end
```

Algorithm 1: After receiving an Interest.

2) After receiving a Data packet: The incoming data name existence is checked in the PIT entries. If no matching entry is found, the data packet will be dropped. Whenever a PIT entry matching the data handled by the ant is found, AC-QoS-FS starts by releasing the reserved bandwidth from the incoming interface of the ant, then it computes the RTT and updates the residual bandwidth and cost of the path between the current node and the node from where the data was retrieved, as explained in Section 1. Thus, it saves these QoS parameters metrics in the fields corresponding to the incoming interface of the data in the FIB entry matching the incoming data. After that, AC-QoS-FS updates the QoS parameters metrics handled by the ant about the path traversed, i.e., the path cost and path residual bandwidth, which are the same values that were saved previously. Next, it computes the pheromone amount and deposits (saves the values) it in the field dedicated for this goal as explained in Section II-A5. Finally, AC-QoS-FS sends a copy of the incoming data (ant) via all the interfaces of the requesting interfaces list to the nodes on downstream, and stores the received data in CS. Once the data is forwarded to downstream nodes, it deletes its PIT entry. Algorithm 2 details how our forwarding strategy works when a router receives a Data packet.

```
if ((Data.name) is found in PIT) then
    RTT(X,P,S) = Time_Receiving − Time_Forwarding
    CostP(X,P,S) = Ant.Cost() + Cost(X,Y,S)
    |BW(X)_int−S−res| = |BW(X)_int−S−res| + |BW_req|
    |BW(X,P,S)_int−S−path| = min(|BW(X)_int−S−res|, Ant.BW())
    Ant.BW(|BW(X)_int−S−path|)
    Ant.Cost(CostP)
    Update the pheromone using Equations 10 and 12
    CS.insert(Data)
    for (each face K ∈ Requesting Faces) do
        Forward(Ant, K)
    end
    Erase(PIT.entry)
else
    Drop data
end
```

Algorithm 2: After receiving a data packet.

III. Performance Evaluation

To evaluate the proposed forwarding strategy AC-QoS-FS, we have used the ndnSIM 2.0 simulator [3]. The performances of AC-QoS-FS are compared with the Best Route forwarding strategy [4]. Our analysis is based on the data delivery time (time elapsed between the sending of Interest and data reception), the cost (the sum of links cost traversed by the data packet from the data source to the consumer) and the hop count (the number of hops traversed by the data between data source and the consumer). We also evaluate in terms of the dropped packets and the mean hit ratio. We have used the Abilene topology as a core of the simulated network, which is composed of 12 routers. In the simulation scenario, each core router is attached to 1 repository used to store permanent contents and a variable number of consumers between 2 and 4. Thus, the nodes total number is equal to 64. We have used a catalog of 100000 contents, where each content has 3 randomly replicas in the different repositories and each router can store up to 1000 chunks in its CS. The consumers request contents according to the largely adopted Zipf's law, concerning the contents popularity, for the parameter values of $\alpha = 0.8; 1.0; 1.2$ [5] with different frequencies: 10, 20, and 30 Interests per second.

The AC-QoS-FS results depicted in Figures 3a, 3b and 3c have been obtained with the following configuration: $\alpha = \beta = \gamma = 1/3$ which means that RTT, bandwidth and cost have the same importance for the application or the end user and we have tuned $\rho = 0.2$ and $Q = 100$. Figure 3a compares the mean cost associated with data transfers using our strategy and Best Route. For a frequency of 10 requests per second, our strategy is roughly $8\% - 10\%$ higher than Best Route, for the three values of the Zipf-law α parameter. However, there is an interesting trend showing that, the higher the frequency of requests, the lesser the cost of AC-QoS-FS. In the case of a frequency equals to 30, the reduction of the costs remains around 10% for the 3 values of α. This shows that for scenarios with increasing loads (several requests flowing), our strategy tends to cost less.

Figure 3b shows that the hit ratio for our solution is, in general, less important than the other strategy. As a result, the average number of hops an ant must perform is slightly higher for AC-QoS-FS, as seen in Figure 3c. This phenomenon is explained due to the nature itself of our algorithms, which is based on a random selection of the path to be followed by the ant, according to the amounts of pheromone at each router. This implies that, a path leading to a next hop with low likelihood of hit ratio can be also selected. However, our results show that this behavior has no significant impact on the average round trip time, since it remains roughly identical for both solutions. Also important, the rate of dropped packets is similar, even if the ants perform longer hops to retrieve information (both round trip time and drop ratio have less than 1% of difference between algorithms). By lack of space, such results are omitted.

The results of different scenarios, that we have executed, demonstrate that we can design a forwarding strategy that could ensure the best performances by tuning ($\alpha, \beta, \gamma, \rho$ and Q) parameters according to the application or end user requirements.

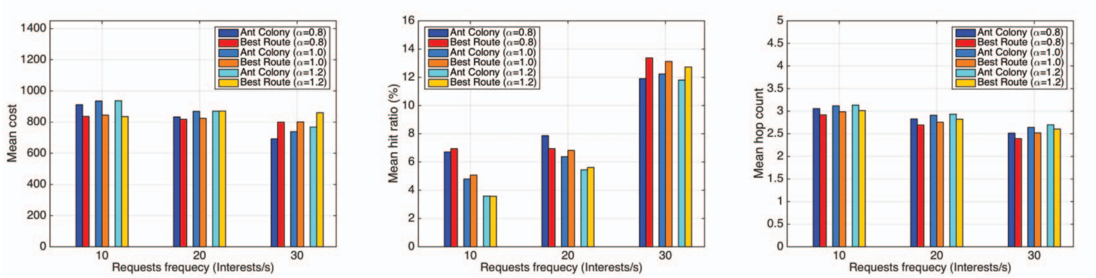
IV. Related Work

The forwarding strategy is the key module for the NDN node architecture. The routing process success depends essentially on the forwarding strategy. The most known forwarding strategy are integrated by default in the NDN simulator [4] and [3]. These strategies are based on the Best Route algorithm, where the interfaces are classified according to number of corrected Data packets retrieved through them.

Afanasyev et al. [6] propose a new forwarding strategy that handles the problems of prefix hijack, link failure, and congestion. The main idea is to use the Interest *Nack* as a tool for the first time, and color-coding (Green, Yellow, Red) to represent the working status of each interface. Chiocchetti et al. [7] propose *INFORM*, a distributed on-line request forwarding algorithm based on Q-routing. *INFORM* discovers temporary copies of content not addressed in routing tables and forwards requests over the best performing interface at every hop.

Authors in [8] compared two approaches: a deterministic exploitation of forwarding information toward a known copy, and a random network exploration toward an unknown copy, via request flooding. They suggest a hybrid solution: trade-offs of exploitation and exploration approaches. Wang et al. [9] propose a hop-by-hop Interest shaping algorithm that takes into account the unique interdependence between Interests and data in NDN to achieves proportional fairness between two-way traffic in order to use efficiently the available bandwidth and avoid congestion.

Authors in [10] present *SoCCeR-Services* over Content-Centric Routing, that extends CCN with integrated support for service routing decisions leveraging ant-colony optimization. The authors in [11] propose a probabilistic ant-routing mechanism that enables multipath transmissions for CCN nodes. In [12], [13], the authors propose a Greedy Ant Colony Forwarding (GACF) algorithm which uses the ISP-based aggregation to reduce the content naming space, inspired from AntNet [14]. This approach suffers from the big traffic generated only for the exploration. In [15], the authors propose probability-based adaptive forwarding strategy



(a) Mean cost (b) Hit ratio (c) Mean hop count

Fig. 3: Obtained results.

(PAF) in NDN by customizing ant colony optimization and focuses on delay minimization. The authors in [16] propose SAF imitates a self-adjusting water pipe system, which guides and distributes intelligently Interests through network crossings circumventing link failures and bottlenecks. A more recent work [17] uses the ant colony algorithms for obtaining QoS. However, the impact of each parameter for the pheromone update (cost, bandwidth, delay) is weighed separately.

In [18] a Parallel Multi-Path Forwarding Strategy (PMP-FS) is proposed, which proactively splits traffic by determining how the multiple routes will be used. In [19] an adaptive forwarding strategy with QoS for NDN is proposed, where each node of the network monitors, in real-time, ingoing and outgoing links to estimate the QoS parameters to determine when and which interface to use to forward an Interest.

In our Performance Evaluation, we were unfortunately unable to include in this evaluation other competitive algorithms (e.g. PAF [15] and GACF [12]) besides Best Route, since we had insufficient details for a decent implementation.

V. Conclusion

This paper presented the **A**nt **C**olony based **Q**uality-of-**S**ervice (QoS)-aware **F**orwarding **S**trategy (AC-QoS-FS) designed to solve the issue of routing with QoS in NDN. It is a new NDN adaptive QoS-aware strategy that takes into account the similarities between the natural behavior of real ants and NDN forwarding process. We have considered three QoS parameters: RTT, bandwidth, and cost. The performance evaluation of AC-QoS-FS in different simulation scenarios shows the effectiveness of our solution. Furthermore, the obtained results demonstrate that we could design a forwarding strategy that ensure the best performances by tuning ($\alpha, \beta, \gamma, \rho$ and Q) parameters according to the application or end user requirements.

In the future, we plan to investigate other QoS parameters that are related to congestion issues in NDN networks.

References

[1] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, and G. Polyzos, "A survey of information-centric networking research," *Communications Surveys Tutorials, IEEE*, vol. 16, pp. 1024–1049, Second 2014.

[2] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *CoNEXT '09*, pp. 1–12, ACM, 2009.

[3] I. M. Alexander Afanasyev and L. Zhang, "ndnsim: Ndn simulator for ns-3," in *NDN, Technical Report NDN-0005, Revision 2: October 5, 2012*, 2012.

[4] I. M. Spyridon Mastorakis, Alexander Afanasyev and L. A. Lixia Zhang University of California, "ndnsim 2.0: A new version of the ndn simulator for ns-3," in *NDN, Technical Report NDN-0028*, 2015.

[5] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, "Impact of traffic mix on caching performance in a content-centric network," in *Computer Communications Workshops, 2012 IEEE Conference on*, pp. 310–315, IEEE, 2012.

[6] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, and L. Zhang, "A case for stateful forwarding plane," *Comput Commun*, vol. 36, pp. 779–791, Apr. 2013.

[7] R. Chiocchetti, D. Perino, G. Carofiglio, D. Rossi, and G. Rossini, "Inform: A dynamic interest forwarding mechanism for information centric networking," in *Proceedings of ICN '13*, pp. 9–14, ACM, 2013.

[8] R. Chiocchetti, D. Rossi, G. Rossini, G. Carofiglio, and D. Perino, "Exploit the known or explore the unknown: Hamlet-like doubts in icn," in *Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking*, ICN 12, pp. 7–12, ACM, 2012.

[9] Y. Wang, *Caching,Routing and Congestion Control in a Future Information-Centric Internet*. PhD thesis, NC State University, 2013.

[10] S. Shanbhag, N. Schwan, I. Rimac, and M. Varvello, "Soccer: Services over content-centric routing," in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, pp. 62–67, ACM, 2011.

[11] J. Eymann and A. Timm-Giel, "Multipath transmission in content centric networking using a probabilistic ant-routing mechanism," in *International Conf. on Mobile Networks and Management*, pp. 45–56, Springer, 2013.

[12] L. Chengming, L. Wenjing, and K. OKAMURA, "A greedy ant colony forwarding algorithm for named data networking," *Proceedings of the Asia-Pacific advanced network*, vol. 34, pp. 17–26, 2013.

[13] C. Li, K. Okamura, and W. Liu, "Ant colony based forwarding method for content-centric networking," in *Advanced Information Networking and Applications Workshops, 27th International Conference on*, pp. 306–311, IEEE, 2013.

[14] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.

[15] H. Qian, R. Ravindran, G.-Q. Wang, and D. Medhi, "Probability-based adaptive forwarding strategy in named data networking," in *2013 IFIP/IEEE International Symposium on Integrated Network Management*, pp. 1094–1101, IEEE, 2013.

[16] D. Posch, B. Rainer, and H. Hellwagner, "Saf: Stochastic adaptive forwarding in named data networking," *IEEE/ACM Transactions on Networking*, 2015.

[17] Q. Huang and F. Luo, "Ant-colony optimization based qos routing in named data networking," *Journal of Computational Methods in Sciences and Engineering*, vol. 16, no. 3, pp. 671–682, 2016.

[18] A. Bouacherine, M. R. Senouci, and B. Merabti, "Parallel multi-path forwarding strategy for named data networking," in *Proceedings of the 13th International Joint Conference on e-Business and Telecommunications*, pp. 36–46, Scitepress, 2016.

[19] A. Kerrouche, M. R. Senouci, and A. Mellouk, "QoS-FS: A new forwarding strategy with QoS for routing in named data networking," in *2016 IEEE International Conference on Communications (ICC)*, Institute of Electrical and Electronics Engineers (IEEE), may 2016.