RankRoute: Efficient Interest Forwarding Using Nodes Ranking

Dongbiao He*, Cedric Westphal †, Jinlei Jiang*, Guangwen Yang*
*Department of Computer Science and Technology, Tsinghua University, Beijing, China
†Department of Computer Engineering, University of California, Santa Cruz, Santa Cruz, CA
{hdb13@mails., jjlei@, ygw@}tsinghua.edu.cn, cedric@soe.ucsc.edu

***Abstract*—The exponential growth of data traffic creates great challenges for content delivery. As a novel paradigm, Information Centric Network (ICN) can better deliver content using Interest aggregation and in-network caching. However, most research has ignored the potential improvement of using paths that are individually sub-optimal between a single client and the content, but offer opportunity to aggregate the traffic from multiple clients along the path. We consider Named Data Networking (NDN) as the basic architecture, and present RankRoute, an Interest forwarding solution that guides the Interests to meet two objectives: 1) allow for Interest aggregation and the creation of multicast opportunities; 2) discover nearby replicas of the content. To achieve this purpose, RankRoute ranks the nodes by their connectivity and the popularity of the workload they serve. Comparisons with several benchmark forwarding strategies show that RankRoute reduces the data transfer completion latency by almost 32%, while the rate of dropped packets is comparable with the ideal Best-Route algorithm.**

***Index Terms*—Content-centric network, cooperative transmission, Interest control.**

I. Introduction

Content distribution has contributed to an exponential growth of the Internet traffic. Concurrently, the Internet has changed its role from mediating communications between remote hosts into connecting users with contents. Information-Centric Networks (ICN) [1], [2] have been suggested to improve the current Internet architecture. ICNs devise a clean-slate architecture and provide in-network caching to enhance content availability. In addition, it provides a name resolution system to naturally support multicast and security. Therefore, it has much greater potential for making the network flexible and intelligent.

In this paper, we consider Named Data Networking (NDN) [3]. There are two kinds of packets in NDN [4], [5]: the *Interest* packet, which is used for seeking the required content along the path constructed by *Forwarding Information Base* (FIB) tables; and the *Data* packet. Data packets are used to carry the content requested by the Interest via the reverse of the path followed by the Interest. This is accomplished by leaving a per-packet state in a *Pending Interest Table* (PIT).

This forwarding plane enables the network to inherently support data packets multicast via coupling data transmission and routing [6], [7], [8]. Hence, these strategies often seek to satisfy a dual objective: to find the nearest replica of the content, as well as to maximize the cache hit ratio.

Interests aggregation is also inherently supported with the NDN forwarding plane. When there are multiple consumers issuing concurrent Interests for the same content, the routers (i.e., the intermediate nodes) on the Interests' path aggregate these Interests. These means that new Interests for a piece of content that has already been requested are not forwarded. However, when the content is being received at this node, it is sent back to all the clients who requested it. Since the content is only sent once by the source, then replicated at the nodes which aggregated the Interests, this creates a multicast tree for the distribution of this piece of content. This multicast mechanism has the advantage of saving bandwidth and reducing the load at the content producers when compared with TCP/IP networks. Recent work [9] considered a scenario where users requested approximately 1.8 million unique files using 5.7 million requests so that two out of three requests were duplicates on average. Their simulation results also illustrate the benefit of Interests aggregation in reducing the network load.

Unfortunately, current Interests forwarding and data transmission strategies [10] mostly neglect Interests aggregation. We argue forwarding should better utilize this feature of NDN to enhance data delivery. We propose RankRoute, a solution that reduces redundant data transmissions while enhancing the content delivery by forwarding Interests towards path whith higher possiblity of aggregation. The key idea of RankRoute is to preferrentially select the nodes which have better connectivity as the pivot node to forward Interests. Our contributions are listed below:

- We build the concept of node ranking and devise an Interest forwarding scheme to select routing paths for data transmission with a low-latency guarantee.
- We evaluate the performance of our method through simulations. The preliminary experimental results show that RankRoute can reduce latency by 32% in a complex network with about 200 nodes. In addition, the low latency forwarding algorithm keeps the packet drop rate similar when compared with the ideal Best-Route policy.

II. Problem and Motivation

NDN utilizes Interest packets in multiple layers to access data. Ideally, it is preferable that the Interests should be delivered to the nearest replica of the content, so as to access the objects with the minimum latency. This is what we denote by Best-route (when routing is unaware of cached content). This is the policy followed by iNRR [7] when caches are taken into consideration. We consider Best-Route in this Section, namely a policy that takes the shortest path towards the nearest origin server for the content.

Best-route is not optimal: Let us look at the example shown in Figure 1. There are two topologies in this figure, both including five nodes. Node A and D are the consumers and we regard E as the producer. In addition, all the links have the same bandwidth: $1Mbps$. The major difference bewteen these two topologies is that topology-b has an extra link BE when compared with topology-a. Intuitively, topology-b has more network resources and therefore should perform better -or at least no worse- in completing data transfers. We apply the "best-route forwarding method to these topologies with all the same parameters. Surprisingly, the consumer A gets a longer completion time when we add the link BE to the topology, as shown in Table I.

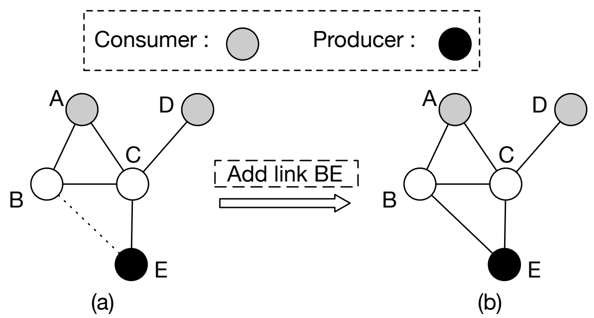


Fig. 1: Sample topology

Item	Without BE	With BE
Packets on E: No Cache	211.7	423.4
Aggregation	800	0
Delay: With Cache	0.0288	0.0576
Cache hit ratio	100%	0%

TABLE I: The simulation result

It so happens that consumers A and D share the link EC to transmit data in topology-a for the data transmitted by node E. If we place the caching capacity in the network, the latency of D is cut by half since D can directly fetch the items from node C. The following points are our main observations:

(1) Lack of information for aggregation opportunities: Best-route only chooses the best next hop for forwarding with the lowest routing cost. It is efficient to find the shortest path to the producers but it neglects the opportunities for building efficient data transmission paths shared with others to take advantage of in-network caching and Interests aggregation.

(2) Lack of information to reach the nearest replica: Since both A and D require the same content, C could be the potential caching relay for the consumers, due to Interests aggregation and/or in-network caching mechanisms. As is displayed in table I, Node A gets 100% cache hits in topology-a while it is always a cache miss in topology-b since A and D get the content from E via different paths.

Aggregation of Interests: in recent years, some works coupled Interests forwarding with caching for routing directly to the cache nodes, such as iNRR [7], CATT [11] and SAF [12]. They focused on routing to the nearest replica to improve the cache utilization as well as reducing the latency for retrieving the data object. In addition, the Interests appended in the PIT would also become a potential content pointers for the consumers. However, the aggregation mechanism has not been well-utilized as [13] showed that the rate of aggregation in the network is low. Despite the presence of in-network caching, the current forwarding methods often neglect the potential aggregation capability of the Interests. For this reason, we investigate a routing algorithm that takes into account Interest aggregation opportunities.

Algorithm 1 An overview of RankRoute

```
1: if received or generated an Interest packet for forwarding then
2:     Obtain the Rank value of next hop neighbors (Equation 1)
3:     Choose the largest Rank Value with low-latency guaranteed algorithm (Equation 2)
4: end if
5: if received or generated a data packet for transmission then
6:     Make caching decisions with RankRoute
7: end if
```

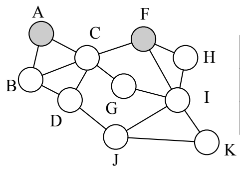
III. Forwarding with RankRoute

This section details our solution (RankRoute) to fulfill the goals proposed in Section II. Generally speaking, RankRoute attempts to detect opportunities for the aggregation of Interests. An overview of RankRoute is displayed in Algorithm 1.

A. Node Ranking Definition

In NDN, the PIT contains the arrival interface of the Interests that have been forwarded, but are still waiting for the matching Data in response. This information is required to deliver data to their consumers. However, if the PIT entries are timed out prematurely, the corresponding Data will be dropped, and the consumer has the responsibility to retransmit the Interest. To maximize the usage of the PIT, a node does not forward a subsequently received interest for data that has already been requested. It instead adds its incoming interface to the existing PIT entry. Since the PIT includes the set of interfaces over which the Interests have arrived, it provides natural support for multicast functionality. Hence, in this section, we introduce the concept of "Node Ranking" to guide the Interest Packets to more efficiently use PIT for aggregation and data transmission.

We start from a special case with the network topology containing two clients, and the definition of the "Node Ranking" could be easily extended to various network states. For two clients n_i and n_j in a given topology, we can list all the



Clients	No links shared	With shared links
A	A,F,B,C	D,G,H,I,J,K
F	A,F,B,C	

Fig. 2: An example of nodes classification

paths from these two clients to the other nodes. With respect to transmission efficiency, we only choose the paths within the distance: $(1+\theta) * L_{sd}$, where L_{sd} represents the distance of the shortest path from node s to the destination d, and θ is a variable that relates to L_{sd} so as to put more paths into consideration. From these paths, we could classify the nodes in the network via the shared links between the two clients. An example is displayed in Figure 2. There, both of the clients A and F have their own non-shared links node set while other nodes are all in the set of nodes which have shared links to the clients. In general, if there are two nodes n_i and n_j, we use $R_I(i)$ to represent the set of nodes of n_i that has no shared links with that of n_j, and use $R_I(j)$ to represent the region of n_j that dose not have overlap in paths with that of n_i. Finally, $R_I(ij)$ is used to to denote the remaining nodes. From the definition, We can see that when n_i is communicating with nodes in the $R_I(i)$, there will be no sharing of links along the packets forwarding paths to the other client n_j. Similarly, it will also happen when node n_j requires the content in the area of $R_I(j)$. We now turn to the relationship between $R_I(j)$ with n_i or $R_I(i)$ with n_j.

Proposition 3.1. For a given connected topology, $R_I(i)$ is the same with $R_I(j)$ for any clients n_i and n_j. It is represented as R_I uniformly.

Proof. Assume the proposition is false. This conditional statement being false means the set $(R_I(i) - R_I(j)) \neq \varnothing$. Thus, there exists at least one node $n' \in V$ for which $n' \in (R_I(i) - R_I(j))$. We assign this node to the set $R_I(i)$, and get paths P_i' from n' to n_i. Since the graph is connected, there exists one or more paths P_j' from n' to n_j. From the assumption and definition, n' also belongs to the set $R_I(j)$ since P_i' and P_j' share no common links. This is a contradiction with the assumption and the proposition is true.

Definition 3.1. The *ranking nodes* of two given clients are the elements of R_I.

If clients n_i and n_j request the items located in the range R_{ij}, they will always go through the pivot nodes, which would offer the possibility of shared links. See the example in Figure 2. If the required items are located in the node J, node A could choose B and C to reach J. As for node F, it has node C and H to choose. Since B and C are the pivot nodes, it can generate the shared paths which are $\{B, D\}$ and $\{C, D\}$ respectively. As a result, the differences in choosing the forwarding nodes to transmitting packets of node n_i and n_j lead to different path sharing and latency of the nodes in the network.

Here, we define the workload $\mathcal{W}(i)$ of node n_i as the accumulated number of packets received by the node through the entire period (either Interests from the clients, or data from the content producers). We use $\mathcal{L}$ to represent the data transmission load in the network for the clients. Owing to the nature of NDN, we will have $\mathcal{L} < \sum \mathcal{W}(i)$ with Interests aggregation and in-network caching. As mentioned before, the major goal of node ranking is to maximize the equation $(\sum \mathcal{W}(i) - \mathcal{L})$. That is to say, we want to transmit the least amount of redundant data to satisfy more requests with the same content.

From the analysis above, we can decouple the workload $\mathcal{W}$ into two part: one is from the content producers to the ranking nodes, which could aggregate request from other clients; the other part is from the ranking nodes to the client nodes via multicast.

Definition 3.2. The *ranking score* of the nodes is proportional to the equation: *(aggregation ratio)/(multicast cost).*

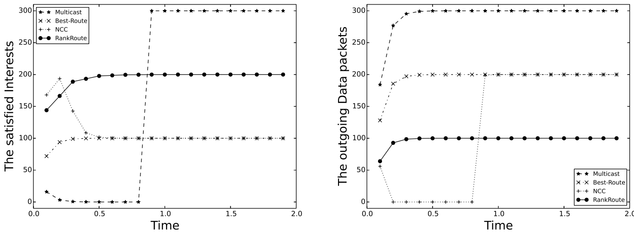
B. Ranking Score Calculation

As discussed, we should identify the nodes that have the potential to disseminate the information to the clients needed via aggregation and caching. In general, finding an exact solution to rank the nodes is not possible since the network is dynamic and changing all the time. The computation and communication costs for this task is unacceptable in practice. From the definition, we can discover that the ranking nodes are always the last hops in the shared transmitting path. They should also be at a short distance from the clients. That is to say, the node with high ranking score should have a good connectivity with the clients. Therefore, we consider network centrality to help ranking nodes. On the other hand, we need to identify the aggregation opportunities of the requests. The popular workloads are more welcome on the paths with higher ranked nodes, while the workloads specific to only a few users are forwarded through the paths with lower ranked nodes. Taking these two factors into consideration, we will have the following definition for calculating the ranking score.

FIB-based node ranking calculation: In principle, the FIB based ranking score calculation of node n_i could be defined as the total number of the out-faces among all the entries $(fib(entry))$. To better describe the connectivity of the nodes, we also consider the node $degree$ into the ranking score calculation. Thus, the equation in this part becomes: $rank_1(i) = \lambda fib(entry) + degree$, where $\lambda > 1$ represents the weight of out-face number and node degree.

This rule is simple to implement at routers. To calculate the rule, each node needs to obtain the number of edges to its neighbors and they should also transmit the result to its neighbors. These computations may be run in the background since some protocols already need this basic information for connectivity.

Workload popularity estimation: In order to promote efficient Interests aggregation and in-network caching, each node calculates the popularity of the workload in a distributed and local way. It has the following benefits: (1) The workload popularity can be estimated in a distributed way; (2) It



(a) The satisfied Interests on C

(b) The out data of node J

Fig. 3: Idealized packets throughput

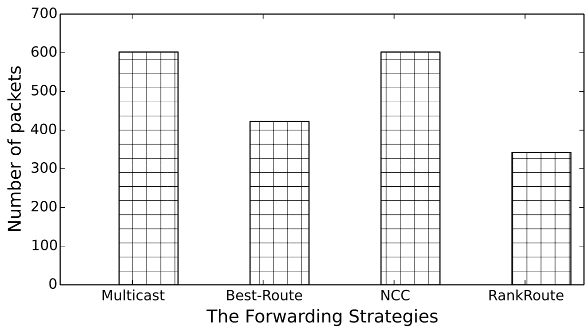


Fig. 4: Total number of packets

matches the actual environment, since consumers in different zone might have their own specific interests.

We use a Count-Min sketch [14] to detect the workload popularity in the node. The Count-Min sketch consists of a set of hash functions. It maps the Interests name to different locations of the functions, and increases the values by one when mapped. Finally, the smallest value among the functions is chosen to be the popularity of the workload ($p(entry)$) in this nodes. Thus, we can get the rank score of workload popularity estimation: $rank_2(i) = p(entry)$.

Ranking Score Calculation: In this paper, we use $1/rank_1(i)$ to estimate the multicast cost and the workload popularity to represent the aggregation ratio. Hence, the score of a node can be calculated according to the Definition 3.2:

$$rank(i) = \frac{rank_2(i)}{1/rank_1(i)} \tag{1}$$

C. Idealized Simulation

To demonstrate RankRoute potential, we simulate an idealized network with no packet loss. We implement it in ndnSIM [15]. For the topology in Figure 2, we choose four consumers (A, B, H, K) with two producers (J, I). The consumers send 100 Interests per second. In this simulation, the node C is selected as the high rank node for A and B by the three methods. In addition, node I is ranked high by node H and K with the view of topology. Hence, we mainly trace the packets transmitted in node C and J.

Figure 3(a) illustrates the satisfied Interests packets of nodes with the four different forwarding algorithms. The amount of Interests satisfied by C increases considerably at the beginning, except for the NCC method. Furthermore, the trends for best-route and RankRoute are very similar while multicast has a sudden increase in 0.8 second. From this figure, we can discover that RankRoute has the twice size of the Interest packets compared with best-route, since both A and B propagate the Interests to C via ranking the nodes while only A forward its interests packets to C with best route. As for multicast, H also spreads its Interest packets to C as well, leading to an increase in of 0.8 seconds.

For a better illustration of the benefit of aggregation, we draw the result of data transmitted from E in Figure 3(b). As we can see, RankRoute, best-route and multicast have similar trends as time grows. The node J transmitted almost nothing at the beginning in NCC. From this figure, node J only transmits once for the required content, since the Interests from A and B are aggregated in C when RankRoute is applied. Best-route uses different paths to transmit the data for each consumer and multicast utilizes all the possible paths for data transmissions. As we have mentioned before, this could cause data redundancy in the network, causing an inefficient utilization of the network resources. The Figure illustrates that RankRoute reduces the load for the content producer.

Figure 4 compares the number of packets transmitted in the network with four different forwarding strategies within 2 seconds. The transmission with node ranking has the smallest amount of packets dissemination in the network, in terms of all kinds of packets, throughout the period shown. NCC and Multicast transmit 602 packets in the network for obtaining the data while best-route needs 402 packets, and RankRoute even fewer. Actually, excluding NCC, the other three methods transmit almost the same amount of data in each links. The number of the links needed in this topology for the three methods is: $8, 6, 5$. The ratio is almost the same as the results in Figure 4.

D. Low Latency Guaranteed forwarding Algorithm

As illustrated previously, we select the nodes with high rank in the network and forward the Interest packets to these nodes with different priority. However, this might lead to the worst situation that most of requests burst into one or few node, which would cause some congestion and degrade the network performance. Since the goal of short delay is the major real-time requirement for traffic transmission, we could not directly and simply use the node ranking score to forwarding Interests in practice. Thus, we propose the *Low Latency Guaranteed Forwarding Algorithm* in this section to solve the problem.

Consider a set of nodes $n_1, n_2, ..., n_m$ which are trying to choose the node n_i as the next hop for forwarding the Interest packets. If node n_i has cached the Interest, we can allocate the Interests flow to n_i directly. Otherwise, the algorithm being displayed would give a guidance whether it is suitable to add a new PIT entry for the node n_i. Let us use S_i to denote the size of the packet steam from node n_i, use W_i to denote the bandwidth of node i, and T_i to denote the packet arrival interval from node n_i. The node n_i

with neighbors $n_1, n_2, ...n_m$, should be satisfied with the the following expression according to the scheduling model [16].

$$W_i \geq \sum_{j=1}^{m} \frac{S_j}{T_j} \tag{2}$$

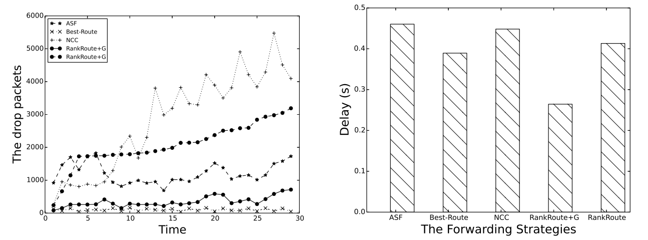
If the equation cannot be solved, it means that node n_i is not able to accept all the Interests. Then some nodes should be prohibited from sending Interests to node n_i. Otherwise, it could wait for the extra space of the node n_i. How to determine the nodes to be prohibited is another important issue in the design.

The ranking score helps us to choose the path which could achieve better path sharing. On the other hand, a node with high rank score has a higher probability to connect with more nodes in the network. For example, as we use the degree centrality, the node that has high degree value would have more candidate nodes to forward the Interests and receive the data. That is to say, we can reallocate these nodes more easily when the bandwidth of the best ranked next hop node is limited.

For example, suppose the bandwidth W_i of the intermediate node n_i is $60kb/s$, the threshold of the workload is 80% of the overall space utility, which is $48kb/s$. Node n_i would estimate the data packet traffic of the three different sources nodes n_1, n_2, and n_3. The estimated data packet size of traffic to n_1, n_2 and n_3 are $10kb, 10kb$, and $20kb$ with arrival interval $0.1, 0.5$ and $1s$, respectively. Then, $\sum_{j=1}^{m} \frac{S_j}{T_j} = 60kb/s$, which is more than $48kb/s$. If n_1 has the largest ranking score, n_i would reject the Interest packets from n_1. After that, the required bandwidth of n_i becomes $40kb/s$. Once the bandwidth of the node drops to 40 kb/s because of the heavy Interests load, n_i overall space utility is reduced to $32kb/s$. Then, n_i informs node n_2 to select other nodes to place its Interests.

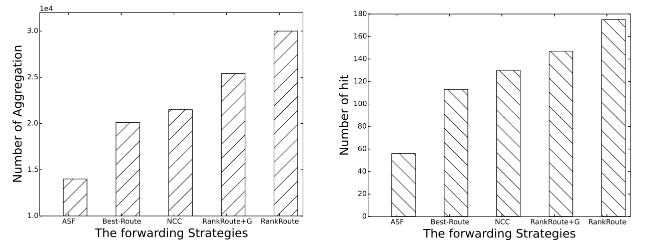
Evaluation result: In this simulation, we test the algorithm in a larger network topology which consists of 191 nodes and 351 links. We randomly choose some of the leaf nodes as consumers, and some of them as producers. The number of consumers is 16 while the number of producers is 4. We assume that the consumer would obtain the items within the giving set $S(|S| = 500)$ at the frequency of 300 request per second.

Figure 5(a) and figure 5(b) display the result of packets dropped and consumer delay when using RankRoute against the performance of ASF[17], Best-route and NCC. We use "RankRoute+G" to represent the RankRoute with low latency guaranteed algorithm. We observed that RankRoute+G and Best-Route has a good performance in our simulation, while others have higher drop rate and delay. From this figure, we can find that Best-Route has the smallest number of the packets dropped, since it always choose the best paths for transmission. RankRoute+G is slightly higher than Best-route, but it drops 25% fewer packets when compared with original RankRoute.



(a) Packets drops (b) Delay

Fig. 5: Performance of the low latency guaranteed algorithm



(a) Aggregation (b) Cache hits

Fig. 6: Performance in aggregation and cache hits

As for delay, RankRoute+G outperforms Best-Route (see Figure 5(b)) since it decreases the amount of redundant data transmission in the network. From the figure, we see that RankRoute+G reduces the latency by 32% compared with Best-Route, and 43% with ASF.

Figure 6 shows the amount of Interests aggregation and cache hits among the different forwarding strategies. We set the cache to content population ratio as $1/30$. As we can see from figure 6(a), RankRoute has the largest amount of Interests aggregated which is almost $1.5\times$ the amount of Best-Route. With the restriction of the bandwidth in RankRoute+G, it reduces the aggregation ratio by 16% to gain in latency. The performance in caching hit times (displayed in Figure 6(b)) is very similar with the Interest aggregation, which implies that RankRoute is also able to have a higher probability to discover the replicas in the cache.

E. Multicast and Caching issues

In a highly dynamic network, transmission links might break down or PIT entries might expire before the data packets arrive. The delay generated in the packet retransmission degrades the network performance and QoS. We can use the allocation algorithm in this paper to make this feature more reliable. As Section 3.4 shows, the space utility of an intermediate node that is used for forwarding a node n_i is $\frac{S_i}{W_i}$. That is, offering more paths for transmitting of the free space can increase the bandwidth utilization of each node (W_i). Based on this rationale, a free space based Interests multicast mechanism could be proposed. The basic idea is that we can send some Interest packets on more paths when there is free space to allocate for the next hop nodes from Equation 2.

In addition, the caching decisions can be applied based on the node ranking. Suppose that at time slot t, node n_i receives the Data Packet containing the first chunk of data object k_{new} not currently cached at node n_i. If there is sufficient unused space in the cache of n_i to accommodate the whole object k_{new}, then n_i proceeds to cache all the Data Packets of k_{new}. That is, the entire data object k is cached at node n_i. Otherwise, the node compares the *previous ranking score* for k_{new} and the currently cached objects. Let C_{old} represent the set of scores of the currently cached items. If there exists $c' \in C_{old}$ with higher score than k_{new}, then c' is evicted and replace with object k_{new}. Otherwise, the cache is unchanged. Data objects evicted from the cache are potentially cached more efficiently and could be reached more easily by the clients somewhere else for it has a larger ranking score.

IV. Related Works

Forwarding Interests properly to efficiently retrieve data content in NDN has attracted a lot of scientific research. In this section, we will give a brief survey of the related work.

[18] showed the benefit of in-network caching in the capacity Information Centric Networks. MAGIC [19] considers caching where there is more benefit to the clients. iNNR [7] uses an iterative algorithm to leverage the availability of content in all caches in the network. This strategy selects the interface with the shortest distance to the content, which prefers nearby caches rather than the content's origin. INFORM [20] is an adaptive hop-by-hop forwarding strategy which uses reinforcement learning inspired by the Q-routing framework. Qian et al [21] proposed the concept of Probability-based Adaptive Forwarding. The basic idea is to select interfaces based on a probability distribution, which is also similar to our approach.

Yeh et al [6] proposed VIP, a framework for joint dynamic forwarding and caching in NDN. Virtual Interest Packets (VIPs) capture the measured demand for respective data objects. The VIP count in a part of a network represents the local level of interest in a given object. The VIP framework employs a virtual control plane which operates on the VIPs. Distributed control algorithms are used to guide caching and forwarding strategies. SAF [12] imitates a water pipe system, intelligently guiding and distributing Interests through the network, circumventing link failures and bottlenecks.

[22] and [23] considered centrality as well to assist with in-network caching, but rather defined some content-specific centrality measures and cache management policies.

V. Conclusion

We have presented the design and evaluation of RankRoute, a solution that enhances content delivery in NDN by finding common delivery paths for the content. We introduced the concept of ranking nodes and build paths to transmit the corresponding data packets towards the nodes with high rank. We also use a low latency guaranteed forwarding algorithm to avoid overloading the paths with high rank nodes. Multicast mechanisms and caching placement policy could also benefit from the ranking score. Our initial experimental results show that RankRoute reduces latency by 32% for data transmission. It ensures low latency and high throughput by finding more sharable paths. In addition, since we implement the RankRoute in a fully distributed way, only minor modifications are needed for deployment in NDN routers.

References

[1] B. Ahlgren et al, "A survey of information-centric networking," *Communications Magazine, IEEE*, vol. 50, no. 7, pp. 26–36, 2012.

[2] G. Xylomenos et al, "A survey of information-centric networking research," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, 2014.

[3] L. Zhang et al, "Named Data Networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[4] C. Yi et al, "A case for stateful forwarding plane," *Computer Communications*, vol. 36, no. 7, 2013.

[5] ——, "Adaptive forwarding in Named Data Networking," *ACM SIGCOMM computer communication review*, vol. 42, no. 3, 2012.

[6] E. Yeh at al, "VIP: A framework for joint dynamic forwarding and caching in named data networks," in *ACM Conference on Information-Centric Networking*, 2014.

[7] G. Rossini and D. Rossi, "Coupling caching and forwarding: Benefits, analysis, and implementation," in *1st international conference on Information-centric networking*. ACM, 2014.

[8] S. Ioannidis and E. Yeh, "Jointly optimal routing and caching for arbitrary network topologies," *ACM ICN*, 2017.

[9] S. Shannigrahi, C. Fan, and C. Papadopoulos, "Request aggregation, caching, and forwarding strategies for improving large climate data distribution with NDN: a case study," in *4th ACM Conference on Information-Centric Networking*, 2017, pp. 54–65.

[10] Q. Chen, R. Xie, F. R. Yu, J. Liu, T. Huang, and Y. Liu, "Transport control strategies in named data networking: A survey," *IEEE Communications Surveys & Tutorials*, no. 3, pp. 2052–2083, 2016.

[11] S. Eum et al, "Catt: potential based routing with content caching for ICN," in *2nd ICN workshop on Information-centric networking*, 2012.

[12] D. Posch, B. Rainer, and H. Hellwagner, "SAF: Stochastic adaptive forwarding in named data networking," *IEEE/ACM Transactions on Networking*, vol. 25, no. 2, pp. 1089–1102, 2017.

[13] A. Dabirmoghaddam, M. Dehghan, and J. Garcia-Luna-Aceves, "Characterizing interest aggregation in content-centric networks," in *IEEE/IFIP Networking Conference*, 2016.

[14] G. Cormode and S. Muthukrishnan, "An improved data stream summary: the count-min sketch and its applications," *Journal of Algorithms*, vol. 55, no. 1, pp. 58–75, 2005.

[15] S. Mastorakis et al, "ndnSIM 2.0: A new version of the NDN simulator for NS-3," *NDN, Technical Report NDN-0028*, 2015.

[16] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of the ACM (JACM)*, vol. 20, no. 1, pp. 46–61, 1973.

[17] V. Lehman at al, "An experimental investigation of hyperbolic routing with a smart forwarding plane in NDN," in *IEEE/ACM Symposium on Quality of Service (IWQoS)*, 2016.

[18] B. Azimdoost at al, "Fundamental limits on throughput capacity in information-centric networks," *IEEE Transactions on Communications*, 2016.

[19] J. Ren et al, "Magic: a distributed max-gain in-network caching strategy in information-centric networks," in *IEEE INFOCOM workshop on Emerging Name-Oriented Mobile Networking*, 2014.

[20] R. Chiocchetti et al, "INFORM: a dynamic interest forwarding mechanism for information centric networking," in *3rd ACM SIGCOMM workshop on Information-centric networking*, 2013.

[21] H. Qian et al, "Probability-based adaptive forwarding strategy in named data networking," in *IFIP/IEEE Symposium on Integrated Network Management (IM 2013)*.

[22] J. Khan, C. Westphal, J. Garcia-Luna-Aceves, and Y. Ghamri-Doudane, "Nice: Network-oriented information-centric centrality for efficiency in cache management," in *ACM ICN*, Sep. 2018.

[23] J. Khan, C. Westphal, and Y. Ghamri-Doudane, "A popularity-aware centrality metric for content placement in information centric networks," in *IEEE ICNC*, Mar. 2018.