2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)

Multipath Forwarding Strategies and SDN Control for Named Data Networking

Mohammad Alhowaidi, Deepak Nadig, Byrav Ramamurthy, Brian Bockelman, and David Swanson
Dept. of Computer Science & Engg, University of Nebraska-Lincoln, Lincoln, NE 68588-0115, USA.
Email: {malhowai,deepaknadig,byrav,bbockelm,dswanson}@cse.unl.edu

***Abstract*—Named Data Networking (NDN) proposes a content-centric rather than a host-centric approach to data retrieval. Data packets with unique and immutable names are retrieved from a content store (CS) using *Interest* packets. The current NDN architecture relies on forwarding strategies that are dependent upon *on-path* caching and is therefore inefficient. This approach reduces data transfer efficiency by ignoring the cached content available on the adjacent *off-path* routers in the network. In this paper, we propose a novel *distributed* multipath (D-MP) forwarding strategy and enhancements to the NDN Interest forwarding pipeline. Furthermore, we develop a *centralized* SDN-enabled control for the multipath forwarding strategy (S-MP) that distributes Interests efficiently by using the global knowledge of the NDN network states. We perform extensive evaluations of our proposed methods on an at-scale WAN testbed spanning six geographically separated sites. Our solutions outperform the existing NDN forwarding strategies by a significant margin. We show that the D-MP strategy results in performance gains ranging between 10.4x to 12.5x over the default NDN implementation without *in-network* caching, and gains of 12.2x to 18.4x with *in-network* caching. In addition, for the S-MP case, we demonstrate a performance improvement of 10.6x to 12.6x, and 12.9x to 18.5x, for with- and without *in-network* caching respectively.**

I. Introduction

In recent years several Internet architectures have been proposed as alternatives to the current TCP/IP Internet architecture. These architectures propose solutions to handle the well-known limitations of the narrow-waist problem of an IP-based Internet. Named Data Networking (NDN) [1] is one such architecture that proposes the use of names for fetching data instead of relying on addresses for identifying data locality. The end-user sends an *Interest* packet with the data name and the network is responsible for both forwarding and caching the requested data. One of the main characteristics of NDN is *in-network* caching, where the router keeps a copy of the data to satisfy a future request. This reduces the latency arising from fetching the data from the source for all subsequent requests. The software defined networking [2] paradigm has generated significant interest in the information centric networking (ICN) community. SDN has been used to address the name-based routing and forwarding [3], [4] by decoupling the ICN data plane from its control plane [5].

Under the NDN paradigm, a content store (CS) acts as a cache management data structure. The CS is an *in-network cache* and performs data lookups for incoming Interests and serves the consumers without the need for forwarding the Interests to the NDN producers. In the current NDN implementation, it is only beneficial to cache the data in the CS when the cached contents are available *on the path* to the content producer. This a serious limitation as it reduces the data transfer efficiency by ignoring the (requested) cached content available on adjacent/off-path routers in the network. The adjacent/off-path routers are generally closer (in terms of the number of hops or routing cost) to the consumer when compared to the NDN producer. Therefore, fetching the data from the producer and caching it only in the on-path router instead of also utilizing the adjacent/off-path routers is inefficient.

In this paper, we propose a multipath forwarding strategy to address the above problem. Our first approach proposes enhancements to the existing NDN Forwarding Daemon (NFD) implementation. Specifically, we propose a forwarding strategy that retrieves non-overlapping data packets from multiple routers simultaneously. Further, the strategy provides additional flexibility in the per-router choice of the Interest pipeline depth configuration. Next, we propose a centralized approach using a SDN controller for managing/mapping the current contents of the CS. This approach allows us to make intelligent Interest pipeline forwarding decisions by analyzing the global view of the NDN network. The SDN controller is effectively used to analyze the network state and redirects the incoming Interests to the off-path routers that have cached the requested content. In our work, we enhance the data retrieval process for both cases by allowing both the NDN consumer and the NDN routers to fetch the content from multiple off-path locations based on the network states. Our proposed approaches, while improving data transfer performance on the one hand, also ensures congestion avoidance on a specific path by distributing Interests across multiple available paths.

The main contributions of our work are as follows: i) we propose a *distributed* multipath (D-MP) forwarding strategy for Interest pipeline processing and data retrieval. This approach enables simultaneous data retrieval from a set of n routers with pre-configured Interest pipeline depths. The D-MP strategy provides a performance improvement of over 10x when compared to the default NDN implementation; ii) we propose a *centralized* SDN-enabled control for our multipath forwarding strategy (S-MP). We show that the centralized control (S-MP), unlike the D-MP case, provides additional benefits due to the knowledge of the global NDN network and cache states; iii) we propose NFD configuration algorithms for D-MP and S-MP approaches, and iv) we evaluate the performance of our solutions on an at-scale network research

978-1-5386-8134-3/18/$31.00 ©2018 IEEE

testbed and provide valuable WAN performance insights.

The paper is organized as follows: Section II presents our system architectures for both the distributed and the centralized SDN-control forwarding strategies, respectively. Section III outlines our solution approaches for the Interest pipeline management for both D-MP and S-MP cases. In Section IV, we present our test network setup, the datasets used for evaluating the performance, and we describe our experiments. Section V presents the performance results for the proposed D-MP and S-MP strategies. We present the related works in Section VI, and in Section VII, we conclude our work.

II. System Architecture

NDN mitigates IP-networking problems such as IP mobility, network address translation (NAT) traversals, and address space limits by using a name-based packet forwarding and routing scheme. A hierarchical and unbounded namespace also solves the problems associated with IP-networks and ensures communication continuity as the data is no longer bound to the host address. This ensures data mobility and eliminates address-space management in the network. NDN routing is similar to its IP-network counterpart, with longest-prefix matches performed on the data *names* instead of the IP addresses. Each NDN router maintains a forwarding information base (FIB) populated with name prefixes. A name-based routing protocol is used to populate each router with the *name* prefix and the associated interface on which the data can be retrieved.

The NDN consumer (i.e. the data consumer) generates and transmits an Interest packet (with the name information) for the desired data request. Each router handles incoming Interest packet processing and looks up the requested name prefix in its FIB. If the requested data is available, the router sends the data back to the consumer along with the data producer's (i.e. the NDN Producer) signature. If the data is unavailable, the router adds the Interest to a pending interest table (PIT) and forwards the Interest to the next router on the path. Reverse-path tracing of the Interest packet is used to deliver the data to the consumer. For each data packet successfully delivered to the consumer, the NDN routers remove the corresponding Interest entries from their PIT and store the data in their Content Store (CS). The CS serves as a cache for subsequent requests to the same data (or name).

The NDN Forwarding Daemon (NFD) is the NDN network forwarder that implements the forwarding functionality for both Interest and data packets. The NFD manages the PIT, the CS, and the FIB. NFD implements the NDN forwarding strategies and the associated forwarding logic. NFD also provides the *Face* abstraction for interaction with various low-level transports. It manages all information associated with data transfers including the forwarding states, strategy used, pending Interests, and the FIB.

NFD employs a per-namespace forwarding strategy to forward Interests. The strategy choice influences packet forwarding decisions, and plays an important role in fetching the data from a given NDN router. A number of Interest forwarding strategies are available for use by the NFD including best routes, multicast, client control, NCC (implemented from CCNx, i.e. CCN backwards), access router, and adaptive SRTT-based Forwarding (ASF) [6] strategy.

The strategies described above, although sufficient for many existing network environments, do not cater to the necessary performance requirements of large-scale distributed datasets. We develop strategies that are suitable for large-volume distributed data transfers over high-bandwidth, high-delay wide area networks (WANs). The targeted use of the developed strategies and forwarding pipelines are complex and distributed filesystems such as CernVM File System (CVMFS) [7]. High-energy physics (HEP) workflows (e.g. Compact Muon Solenoid (CMS) [8]) are evaluating the use of CVMFS for the distribution of experimental datasets [9] using NDN. Next, we outline two approaches to Interest distribution: i) a distributed multipath (D-MP) strategy and enhancements to the existing NDN Interest pipeline, and ii) A centralized SDN-enabled control for the multipath strategy (S-MP).

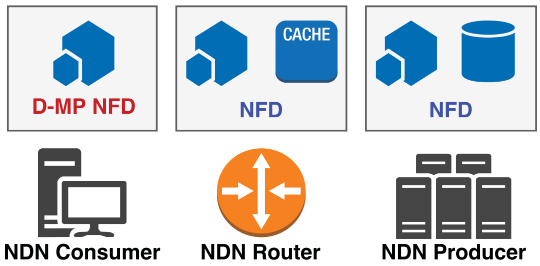


Fig. 1: D-MP Network.

A. *D-MP: Distributed Multipath Forwarding Strategy*

NFD data transfer decisions use a combination of strategy choice and the forwarding pipeline depth. Together, they form the NFDs' intelligence and packet processing logic. The strategy choice influences the packet forwarding decisions. The forwarding pipeline specifies the number of simultaneous Interests that are forwarded per request. We propose NFD strategy enhancements to optimize the NFD forwarding pipeline. The architecture is as shown in Figure 1. Our NFD enhancements enable parallel data (or namespace) retrieval from multiple routers using a per-router Interest pipeline depth. Our proposed distributed multipath (D-MP) strategy benefits from multipath gains and/or *off-path caching* to reduce the latency of data-delivery to the consumer. Unlike the multicast strategy, our optimizations focus on parallel data retrieval from a set of NDN routers. We also consider the effects of caching at the content store (CS) of each router on data retrieval times. Using the D-MP strategy, the data consumer can simultaneously request non-overlapping data segments from multiple routers. Further, our approach enables the consumer to specify a per-router forwarding pipeline depth. Thus, D-MP can optimize parallel data transfers from multiple NDN routers based on the exchanged information between the NDN consumer and NDN routers. Details of the D-MP approach are presented in Section III-A

B. S-MP: Centralized SDN control for the Multipath Forwarding Strategy

The D-MP approach described in the previous section relies on a *priori* information about the available router forwarding paths for forwarding decisions. Although the D-MP approach benefits from multipath data retrieval and larger optimized Interest pipelines per path, it is vulnerable to dynamic network state changes due to its dependence on a priori information. To facilitate the use of real-time NDN network state information in Interest forwarding decisions, we propose a software-defined control architecture for the multipath forwarding strategy. The architecture is shown in Figure 2. The S-MP architecture uses representational state transfer (REST) application programming interfaces (APIs) for information exchange between the NDN and the SDN infrastructures. The centralized SDN control manages the NDN network state information including router states, available forwarding paths, and cached contents. Further, the SDN controller asynchronously communicates with the NDN routers and the content producers to create a data map of the CS. This establishes a global view of the data maps on the NDN routers, and are representative of the data cached in the memory buffer of each NDN router. The S-MP strategy involves the following:

1) Consumer NFD Configuration: First, we choose the set of routers that already cache the requested content either partially or fully. Based on the caching information, we formulate a multi-router Interest distribution strategy and communicate it to the data consumer NFD. The consumer's NFD configures the associated faces, routes and Interest pipeline depths and initiates the parallel non overlapping data retrieval from multiple NDN routers. If the requested data caching is unavailable at the routers, the SDN controller sends a list of the best candidate routers and the corresponding Interest distribution strategy to the consumer NFD. The consumer uses this information as before to set up the connections.

2) Data Retrieval: The consumer NFD establishes parallel connections with the specified list of routers and configures each connection with an associated Interest pipeline depth. Parallel connections retrieve the requested data and assemble it at the consumer. If no data is cached at the routers, the consumer NFD sends the Interests for non-overlapping data segments to the routers. The routers in-turn fetch and cache the requested data from the NDN producers and deliver it to the consumer.

Two types of messages are exchanged between NDN routers and the SDN controller namely, i) *Forward* message, used when the requested data is unavailable at the router, and ii) *Update* message, used when the new data is received and cached at the router. When the Interest arrives at the NDN router, the NFD checks the content store (CS) for the requested data. If the data is available, the router will return a copy to the consumer. Otherwise, the router's NFD forwards the Interest along the path to the NDN producer. The SDN controller uses the *Update* message to update its NDN state information data. Details of the S-MP strategy are described in Section III-B

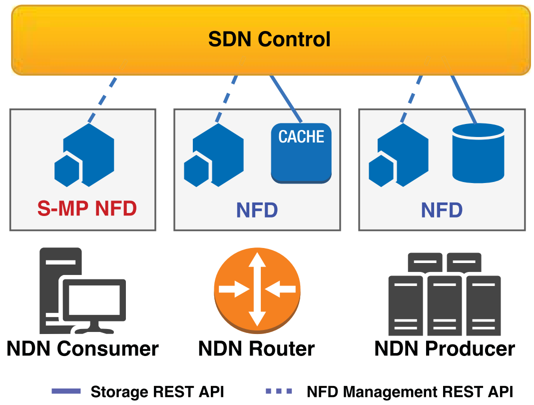


Fig. 2: S-MP: SDN-enabled control for multipath forwarding.

III. Proposed Forwarding Strategies

In this section, we outline different implementation approaches for the Interest pipeline distribution. We discuss our forwarding strategies for both cases, i.e., D-MP and S-MP.

A. Interest Pipeline Distribution Approaches for D-MP

The D-MP-based NFD configuration and data transfers are outlined in Algorithm 1. For each incoming Interest packet, the consumer NFD computes the optimal forwarding strategy, a list of routers and their corresponding pipeline depths. This router set is created based on a discovery phase. During this phase, the consumer sends a message to all routers to check if the data is available in the routers' caches. The consumer processes the replies from the routers and builds a forwarding strategy configuration. This configuration contains information about the NFD Face to use with each router and a per-router Interest pipeline depth.

Different approaches can be used for Interest pipeline distribution for the D-MP case. We implement a round-robin scheme for Interest distribution among a set of n routers. In this approach, we distribute $i, \forall i \in \{1, .., p\}$ *Interests* to the processing pipeline of n routers. This approach ensures that the Interest processing pipeline of each router is always saturated with the defined pipeline depth (i.e. p) for optimal performance. Another approach is to use a ratio-based Interest distribution scheme. In this approach, Interests are distributed to a set of n routers based on a defined ratio partitioning scheme. For example, for a set of $n = 3$ routers and a pipeline depth, $p = 20$, an Interest distribution ratio of $40\%, 40\%$, and 20%, results in the Algorithm 1 assigning 8 Interests for routers 1 and 2, and 4 Interests to router 3 respectively in a non-overlapping fashion. This ratio can be calculated based on the network state information obtained during the discovery phase, or can be changed during the data transfer. Thus, the ratio partitioning approach provides additional flexibility for adjusting the Interest processing pipeline depths for each router to better balance routers' loads and/or processing capacities. In this work we only present the performance results for the round-robin scheme.

Algorithm 1 D-MP(i)

Input: NDN Interest (i).
Output: D-MP NFD configuration for data transfer.
Consumer NFD Configuration Update:
for all $r \in R$ **do**
  Consumer sends discovery message to router r
  Consumer processes reply from router r
end for
Consumer computes optimal *namespace* configuration
Consumer update $C : i_{faces}, C : i_{pipeline}, C : i_{distribution}$
Consumer NFD Data Retrieval:
for all Routers $r \in C : i_{faces}$ **do**
  Configure router pipeline $r_p \leftarrow C : i_{pipeline}(r)$
  Forward *Interests* based on $C : i_{distribution}$
end for

B. SDN-enabled Centralized Interest Pipeline (S-MP)

The SDN-enabled centralized Interest pipeline distribution approach forwards consumer Interests for all requested names to the SDN controller. Algorithm 2 describes the consumer NFD processes and Algorithm 3 describes the SDN controller functionality.

Algorithm 2 S-MP(i)

Input: NDN Interest (i).
Output: S-MP NFD configuration for data transfer.
SDN-enabled Consumer NFD Configuration:
SD-NFDConfig(i)
$C : i_{faces} \leftarrow router\_list$
$C : i_{pipeline} \leftarrow p$
$C : i_{distribution} \leftarrow distribution\_map$
Consumer NFD Data Retrieval:
for all Routers $r \in C : i_{faces}$ **do**
  Configure router pipeline $r_p \leftarrow C : i_{pipeline}(r)$
  Forward *Interests* based on $C : i_{distribution}$
end for

The SDN controller is responsible for computing the forwarding strategy, multi-router configuration and specifying the per-router pipeline depth for a given data transfer request. The SDN controller manages a map of the state of the content store (CS) compiled from all the NDN routers in the network. It also maps the off-path routers that host the requested data. Further, it computes decision statistics based on the routers' status and their network state information. It will then communicate the appropriate strategy configuration to the NDN consumer to help set up the necessary connections. It is to be noted that the communication between the SDN controller and the NDN routers/NDN producers are independent of the consumers' data requests. For every CS state change, the router sends a REST POST to the controller to notify the cache update. Optimum Interest pipeline distribution decisions are made based on the state of the CS of each router (either on- or off-path) in the network. The optimum decision specifies the forwarding strategy, the total number of routers and the associated pipeline depths for configuring the consumer NFD for each router in the configuration.

Algorithm 3 SD-NFDConfig(i)

Input: NDN Interest (i).
Output: SD-NFD Configuration File.
Lookup data map for the *namespace* in Interest i
Compute optimal *namespace* configuration
return $\{router\_list, p, distribution\_map\}$

IV. Experiments

In this section, we describe our test network setup, datasets used in the performance evaluation, associated parameters, and the experiment design.

A. Network Setup

Our test network topology is shown in Figure 3. The test network is composed of two NDN consumers, three NDN routers, an NDN producer, and an SDN controller node. Consumer $C1$ is connected to all three routers ($R1, R2$, and $R3$). $C1$ is the main data consumer for all our tests. It implements our NFD forwarding strategies and Interest pipeline distribution approaches. Consumer $C2$ is only connected to $R1$, and its path to the NDN producer is $C1 \rightarrow R1 \rightarrow R2 \rightarrow R3 \rightarrow P$. The consumer $C2$ is *only used* for populating all routers with the same dataset for the tests with *in-network* caching enabled.

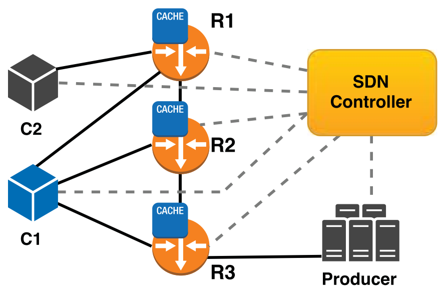


Fig. 3: SDN-enabled MP Test Network.

All nodes in the test network are set up on the GENI [10] platform. GENI provides a platform for at-scale networking research, connecting compute resources over the Internet2 AL2S infrastructure. We use a total of six GENI sites (with one NDN node per site) spread across InstaGENI infrastructures at Georgia Tech, Kansas, Rutgers, Stanford, UCLA and UChicago. Therefore, this setup is representative of a real-world WAN NDN network.

B. Experiments

We evaluate NDN data transfer performance for different scenarios over a WAN test network as outlined in Table I. The two datasets used were: i) 100MB file transfers, and ii) 1000 files of 8KB each. We design the following experiments (*E1* to *E5*) for our evaluations:

1) E1–Single Router, Single Interest Pipeline: This is the default NDN strategy where the consumer retrieves all the requested data from a single router and/or a single producer.

2) E2–Single Router, Aggregate Interest Pipeline: This is a variation of the previous strategy *E1* with an increased Interest pipeline depth ($p = 10$ was used).

3) E3–Distributed Multipath (D-MP), Single-Interest Pipeline: In this case, we use three routers and with only one Interest per router. The list of routers is obtained using the communication between the consumer and the routers. The interest distributed evenly between routers and based on Round-Robin (RR) technique.

4) E4–Distributed Multipath (D-MP) Pipeline: This is Similar to the previous case, but we use multiple Interests per router ($p = 10$).

5) E5 & E6 –SDN-enabled Multipath (S-MP) Pipeline: This is Similar to E3 & E4, except the consumer will retrieve the list of routers which cache the data from the SDN controller.

We evaluate the performance with *in-network* caching both enabled and disabled. NFD manages a content store (CS) which is used to cache the data for a satisfied Interest. For the *in-network* caching-enabled case, the CS at the routers will have cached the requested data and therefore the Interest is not forwarded to the NDN producer. In the default NFD implementation, the interest will be forwarded to the producer and only benefits from on-path caching. In our proposed architecture, the SDN will reconfigure the consumer to send Interests to off-path routers which also host the requested data. Thus, we do not restrict data forwarding only to the on-path routers. This reduces the latency for data retrieval, producer overheads and avoids single-path congestion.

TABLE I: Evaluation Parameters.

Test Datasets	**Experiment Design**		
	#Routers	**Pipeline Size**	**Caching**
100MB Files, and 8KB×1000 Files	1	1	w/ & w/o
	1	10	w/ & w/o
	3	1 per Router	w/ & w/o
	3	10 per Router	w/ & w/o
	3	5:3:2	w/ & w/o

V. Results and Discussion

The WAN data transfer performance of the proposed D-MP and the S-MP methods were evaluated on the GENI network testbed. The SDN controller and all NDN entities (i.e. consumers, routers, and producers) were placed on different InstaGENI sites and aggregated using layer-2 stitching over Internet2 AL2S. Two sets of WAN transfer performance results for two datasets are presented in Figure 4. For both datasets, we evaluate the transfer performance with i) *in-network* caching disabled, i.e. the requested data is not available in the routers' content store (CS), and the requested data is always fetched from the producer and then cached at the router(s); and ii) *in-network* caching enabled, i.e. the requested data is available on both *on-path* and *off-path* routers. All the results in this paper are computed with 95% confidence interval over five runs.

Figures 4a and 4b show the transfer performance results for the experiments listed in the Section IV-B for the 100MB dataset with *in-network* caching disabled and enabled, respectively. Our D-MP approach performs 12.5x and 18.4x better than the default NDN implementation with *in-network* caching disabled and enabled respectively. In addition, the S-MP strategy shows performance gains of 12.6x and 18.5x with *in-network* caching disabled and enabled respectively.

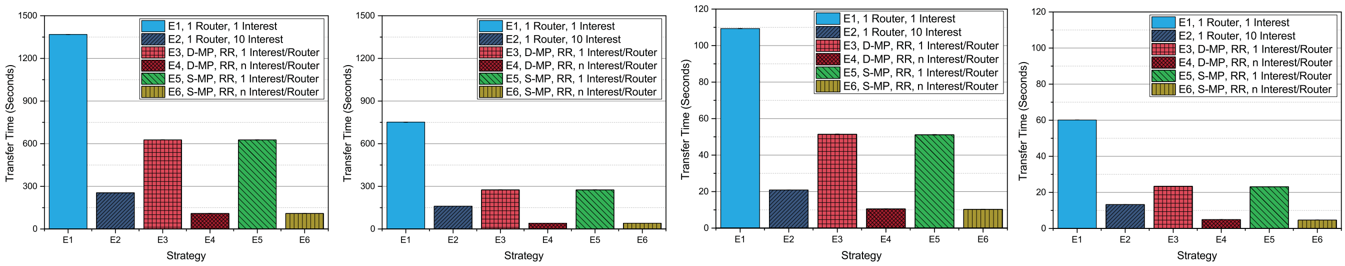
Figure 4c shows the transfer performance for the second dataset (i.e. 1000×8KB files) with *in-network* caching disabled. We see that D-MP and S-MP approaches perform 10.4x and 10.6x better than the default NDN implementation respectively. In Figure 4d, with *in-network* caching enabled, we see further performance improvements, with D-MP and S-MP performing 12.5x and 12.6x better respectively.

Comparing the two proposed approaches, we see that S-MP performs 0.8% and 0.54% better than the D-MP case for transferring 100MB dataset, and it performs 1.92% and 0.8% for transferring 1000×8KB files. The reason for that is that the S-MP only adds a small latency overhead to the transfer time. This is due to the fact that the Interest packet is forwarded to the SDN controller and the consumer waits to receive the configuration update before initiating the connections with the appropriate routers. Furthermore, we note that this is a one-time cost and can be minimized by placing the SDN controller closer to the NDN networks' edge. Thus, the S-MP approach scales predictably with increasing number for Interests. While in the D-MP case, the Consumer needs to contact all routers in the network to build the configuration file which will decrease the performance. The degradation in performance will increase for D-MP case as the number of routers in the network increases as shown in Figure 5.

In order to compare the D-MP strategy with the S-MP strategy, we increase the number of routers by adding another layer of routers to our testbed. The extra three routers are located on three different sites on the GENI testbed and are two hops away from the consumer. Figure 5 shows the communication overhead to build the configuration file for D-MP and S-MP strategies. We observe from the figure that the communication delay increases for the D-MP strategy as the number of routers in the network increases. On the other hand, the S-MP overhead is consistent since it does not depend on the number of the routers, but it mainly depends on the SDN controller's location.

VI. Related work

A number of recent works focus on SDN-NDN integration, improving content caching and placement, and routing/forwarding mechanisms. The authors in [11] proposed the use of a controller to perform content selection and placement on specific off-path routers. Other approaches to optimizing NDN caches include joint-path and off-path cooperative caching policies [12], content popularity based multi-path forwarding and caching strategies [13], and the use of network coding and cache content placement to achieve better bandwidth and cache cost performance [14]. Interest routing and forwarding strategies proposed by the authors in [15] rely on the discovery of temporary copies of content not available in on-path caches



(a) 100MB files, No caching. (b) 100MB files, Caching enabled. (c) 8KB files, No caching. (d) 8KB files, Caching enabled.

Fig. 4: WAN Performance Evaluation of the D-MP and S-MP strategies for different datasets.

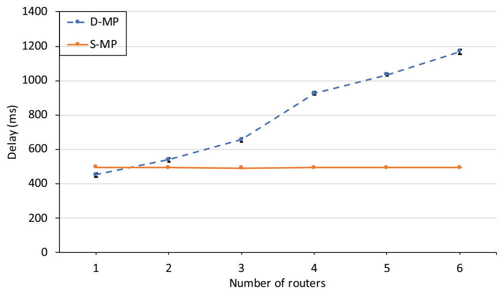


Fig. 5: Comparison of D-MP and S-MP communication overheads.

to forward data requests on each hop. However, the above works focus on caching optimizations to improve existing forwarding strategy performance. The authors in [16] proposed SDCCN, to program CCN forwarding strategies and caching policies using a SDN approach. However, this approach focuses on cache replacement algorithms for improving strategy performance. Unlike the above works, our work focuses on developing novel forwarding strategies for Interest pipeline management.

VII. Conclusions

In this paper, we proposed two novel approaches for Interest pipeline distribution to improve the performance of NDN data transfers. Our D-MP strategy provides up to 18.4x improvement in performance over the current NDN strategies. Next, we presented a centralized SDN-enabled control for the multipath (S-MP) forwarding strategy for Interest pipeline distribution and management. This approach, while providing better flexibility in Interest distribution by creating/managing a map of the current state of the NDN routers' content stores, also provides a 18.5x performance improvement over existing NDN approaches. We evaluate our solutions on an at-scale research testbed to provide valuable insights into the WAN transfer performance of an NDN network. Extensive evaluations with both *in-network* caching enabled and disabled, show that the proposed solutions easily outperform the current alternatives. Finally, the S-MP solution provides a generalized framework for software-defined control of a NDN network. This solution can be easily extended to incorporate both adaptive and intelligent decision making strategies for Interest pipeline management.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Numbers OAC-1541442 and CNS-1817105. This work was completed using the Holland Computing Center of the University of Nebraska, which receives support from the Nebraska Research Initiative.

References

[1] L. Zhang, A. Afanasyev *et al.*, "Named data networking," *ACM SIGCOMM CCR*, vol. 44, no. 3, pp. 66–73, 2014.
[2] B. A. A. Nunes, M. Mendonca *et al.*, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *IEEE Comm. Surveys Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.
[3] S. Gao, Y. Zeng *et al.*, "Scalable area-based hierarchical control plane for software defined information centric networking," in *Intl. Conf. on Computer Communication and Networks (ICCCN)*, Aug 2014, pp. 1–7.
[4] E. Aubry, T. Silverston, and I. Chrisment, "Implementation and Evaluation of a Controller-Based Forwarding Scheme for NDN," in *Advanced Information Networking and Applications*, March 2017, pp. 144–151.
[5] M. Vahlenkamp, F. Schneider, D. Kutscher, and J. Seedorf, "Enabling Information Centric Networking in IP Networks Using SDN," in *SDN for Future Networks and Services (SDN4FNS)*, Nov 2013, pp. 1–6.
[6] V. Lehman, A. Gawande *et al.*, "An experimental investigation of hyperbolic routing with a smart forwarding plane in NDN," in *Intl. Sym. on Quality of Service (IWQoS)*, June 2016, pp. 1–10.
[7] J. Blomer, P. Buncic, and R. Meusel, "The CernVM file system," Technical Report, Tech. Rep., 2013.
[8] S. Chatrchyan *et al.*, "The CMS experiment at the CERN LHC," *JINST*, vol. 3, p. S08004, 2008.
[9] M. Alhowaidi, B. Ramamurthy *et al.*, "The Case for Using Content-Centric Networking for Distributing High-Energy Physics Software," in *ICDCS*, June 2017, pp. 2571–2572.
[10] M. Berman, J. S. Chase, L. Landweber *et al.*, "GENI: A federated testbed for innovative network experiments," *Computer Networks*, vol. 61, pp. 5 – 23, 2014, sI on Future Internet Testbeds - Part I.
[11] H. Salah and T. Strufe, "Comon: An architecture for coordinated caching and cache-aware routing in CCN," in *Consumer Communications and Networking Conference (CCNC)*. IEEE, 2015, pp. 663–670.
[12] H. K. Rath, B. Panigrahi, and A. Simha, "On Cooperative On-Path and Off-Path Caching Policy for Information Centric Networks (ICN)," in *Advanced Information Networking and Applications (AINA), 2016 IEEE 30th International Conference on*. IEEE, 2016, pp. 842–849.
[13] Y. Xin, Y. Li *et al.*, "Content aware multi-path forwarding strategy in Information Centric Networking," in *Computers and Communication (ISCC), 2016 IEEE Symposium on*. IEEE, 2016, pp. 816–823.
[14] J. Wang, J. Ren *et al.*, "A minimum cost cache management framework for information-centric networks with network coding," *Computer Networks*, vol. 110, pp. 1–17, 2016.
[15] R. Chiocchetti, D. Perino *et al.*, "Inform: a dynamic interest forwarding mechanism for information centric networking," in *Proc. 3rd ACM SIGCOMM workshop on ICN*. ACM, 2013, pp. 9–14.
[16] S. Charpinel, C. A. S. Santos, A. B. Vieira *et al.*, "SDCCN: A novel software defined content-centric networking approach," in *Advanced Information Networking and Applications (AINA), 2016 IEEE 30th International Conference on*. IEEE, 2016, pp. 87–94.