CODA: Achieving Multipath Data Transmission in NDN

Dongbiao He*, Jinlei Jiang*‡, Guangwen Yang*, Cedric Westphal†
*Beijing National Research Center for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
†Department of Computer Engineering, University of California, Santa Cruz, Santa Cruz, CA 95064, USA
‡Corresponding author: jjlei@tsinghua.edu.cn
hdb13@mails.tsinghua.edu.cn, ygw@tsinghua.edu.cn, cedric@soe.ucsc.edu

***Abstract*—The exponential growth of data traffic raises a great challenge to content delivery in current TCP/IP networks. To answer this challenge, Information-Centric Networking (ICN) has been proposed with the purpose of bringing content caching and name-based content access to the network layer. Though great progress has been made, most existing ICN proposals lack support for parallel data transfer over multiple paths with low data redundancy. To deal with the issue, we present CODA, a fully distributed cooperative multipath data transmission solution that enhances content delivery further. Taking Named Data Networking (NDN) as a basis, CODA works in a distributed manner with the following contributions: 1) it extends the standard Interest model in NDN to support transmission of data over multiple paths so as to reduce the flow completion time; 2) it devises a traffic scheduling model to form parallel paths for transmitting data in a cooperative way; and 3) it proposes a transmission control scheme to select paths in an efficient and reliable manner. Extensive simulation comparisons with existing data transmission methods show that: 1) CODA speeds up the data rate twice as high as that of the best-route method; and 2) the amount of Interests required by CODA to build multiple data transmission paths in the network accounts for only 66% of that by MSRT, another multipath transmission proposal.**

***Index Terms*—information-centric network, named data network, cooperative transmission, multipath transmission**

I. INTRODUCTION

The growth of Internet traffic will remain unabated in the next few years [1]. Along this trend, the Internet has changed its role from mediating communications between remote hosts into connecting users with contents. However, the current Internet architecture is inefficient in meeting users' requirement for content access [2]. To deal with it, a next generation Internet architecture called Information-Centric Network (ICN) has been proposed [3] and has drawn a lot of attention from both academia and industry. Among many implementations of ICN, we base our work on Named Data Network (NDN) [4] for its popularity and high impact.

A. Motivation

ICN in general, and NDN in particular, functions according to a name based request/response mechanism — a client requests a piece of content by sending an *Interest* message, and receives the desired content in a *data* message. Interest and data are the two types of packets in NDN.

Each router in NDN is equipped with a forwarding information base (FIB) for Interest forwarding. A certain prefix match method is deployed to figure out which interface is used to forward the Interest. If multiple matches are found for the same Interest in the FIB, the router could duplicate the Interest and forward them along multiple paths. However, this is inefficient. Even if the duplicated Interests find multiple copies, all but one of these copies will be dropped at the router that issues the interests, causing a waste of network resource. Moreover, this also creates a risk for distributed denial-of-service (DDoS) and amplification attacks.

For the above reason, NDN currently prefers to retrieve all the chunks of the desired data only along a single path. This is inefficient especially when multiple paths are available to deliver the content from a source node. Further, the data path in NDN is the reverse of the Interest path. This means that the *name resolution* operations also double as the *path selection* operations, making common network functionality such as traffic engineering (TE) difficult to implement, as there is no mechanism to select an alternative path for the data except for the one identified by the Interest.

In the area of Interests forwarding and data transmission in ICN, data transmission along multiple paths could also achieve robustness, congestion reduction and high throughput as is the case in TCP/IP based network. However, most existing methods need to know the locations of the content ahead of time, either by using pre-set routing tables[5] to guide the Interests forwarding intelligently or by aggregating the content sources to the consumers in advance[6], [7], [8] so that the Interest packets can be allocated to different paths. In both cases the routers or consumers should maintain the majority or even all of the network states, which reduces the system throughput when the network conditions are varying over time. Our objective in this work is to construct multiple data transmission paths in a fully distributed manner, while consumers do not have to obtain any knowledge regarding the content sources prior to forwarding the Interest packets.

B. Main Contributions

We propose CODA, a mechanism to enhance NDN with multipath content delivery. CODA leverages a property of NDN, that is, the Interest messages are much smaller than

978-1-5386-6808-5/18/$31.00

the data packets [9], [10]. Therefore, Interests can be used to explore alternative paths. Since the NDN architecture requires that one Interest correspond to one chunk of content, a new type of Interest, namely *Detector*, is introduced to find alternative paths rather than retrieve content.

As a prerequisite for multipath data transmission, CODA is also aware of the properties of chunk-based data transfer in NDN. Since the way the content chunks are delivered does not matter (as long as the chunks reach the consumers in a timely manner, and the content can be reassembled correctly), CODA can employ flexible and efficient schemes to split the content into chunks over multiple sub-flows. We devise a technique to ensure simultaneous sub-flow completion on the consumer side. To achieve this, at the sender side, CODA strategically introduces a delay measurement between the potential sub-flows so that the packets across all sub-flows will arrive at the consumer at the same time. The main function of delay measurement is to estimate the network latency and the available bandwidth dynamically. These measurements have been implemented with Interest packets previously. Doing so simplifies the design. In addition, it decouples the sub-flows and enhances the performance.

In summary, our main contributions are listed below:

- We extend the standard Interest model with a new type of Interest called *Detector* introduced to discover more paths and to construct one primary path and one or more alternative paths for cooperative multipath content transmission;
- We propose a multipath data transmission model using the path information generated by the *Detectors* and devise a distributed traffic scheduling algorithm to split the data transmission into sub-flows with high reliability and high throughput. Meanwhile, we respect the basic NDN principles such as host independence, native support for multicast, and content-location decoupling;
- We evaluate the performance of CODA through extensive simulations. The experimental results show that, in a typical topology, CODA speeds up the data rate twice over the best-route (single path) method. Further, CODA achieves a similar rate but uses only half the amount of Interests when compared with MSRT [7], a method that also supports multipath data transmission in the network.

The rest of this paper is organized as follows. Section II provides some background and points out the design challenges. Section III details the design of cooperative multipath data transmission. Following that is the performance evaluation in Section IV. After a brief overview of the related work in Section V, we conclude the paper in Section VI.

II. BACKGROUND AND CHALLENGES

A. Chunk-based Data Transmission

In NDN, by default, data is delivered to the application on a per-chunk basis. A content chunk consists of a block of bytes specified by the application, which can be, for example, an image part, an audio snippet or a video segment. At the source side, after the producer (cache) server pushes (caches) the data content to the storage buffer, NDN treats all the data in the buffer as multiple chunks by default. This procedure is fully transparent to applications. At the consumer side, when the content is fully received, it is then delivered to the application.

NDN utilizes Interest packets to locate content to implement chunk-based data transmission. The consumer sends Interests for the desired content to the network. Interests are transmitted along the path constructed by FIB tables in the intermediate nodes. If no target data is found locally (in the cache) and no entry exists in the Pending Interest Table (PIT), the node creates a new entry in PIT to store the name of the Interest packet and the interface where the packet is coming from. Then the node forwards the interest packet. If an entry exists in the PIT, the incoming interface of the Interest is added to it, and the Interest is dropped. Otherwise, the corresponding data packets are retrieved and sent back to the consumer along the reverse path that the Interest packet passed through and that had been recorded in PIT.

As long as the desired content can be correctly reassembled, chunks belonging to the content can be delivered in any order. Therefore, the whole content can be split into many parts and each can be distributed onto a different path for delivery. In conventional MPTCP, sub-flows are tightly coupled; a stall (e.g., due to packet loss) in one sub-flow may slow down the other ones due to their limited flexibility in retrieving content. Named data network, on the contrary, decouples the sub-flows by allowing each sub-flow to freely and independently transfer the data until the very end sub-flows are received and merged.

B. Challenges

Ideally, it is desired that the Interests be delivered to multiple sources which have the required content, and these sources would each send a varying-sized part of the content according to their respective network conditions. In this case, the content would be transmitted in a parallel way without much redundant data. This would reduce the transfer completion time, leading to a larger throughput for the network as whole.

Unfortunately, current NDN cannot achieve this goal. To support multipath data transmission in a distributed way, CODA must address the following issues:

(1) Multiple path construction

For constructing multipath transmission in NDN, the consumers need to forward multiple interest packets along different paths guided by the FIBs to find multiple available producers. However, it creates a lot of data redundancy with the multicast, or even the flooding, of Interests. To address this issue, MSRT[7] uses a type of Interest to get the content source locations back to the consumers. However, it could not perform well when the content distribution is changing dynamically. Therefor, a better design should efficiently use the meta data for paths information in real time.

Further, since the control information is related to the size of Interest packets, there is a trade-off between spreading Interests over many paths, and the number of paths constructed for data transmission. Namely, there is a diminishing return in adding more paths, while trying to identify such paths creates an additional overhead.

(2) Cooperative data transmission via multiple paths

On the return path, when new data packets for a specific name reach a node in the network, this node looks up the entries in its PIT which match the name in the data packet. To support multi-path transmissions, multiple corresponding PIT entries are required for the different possible paths, and the data packets must be forwarded to one of the interfaces recorded by these entries. A path selection mechanism is therefore required. Further, to transmit data cooperatively, an efficient mechanism for splitting traffic over the multiple paths should be proposed to transmit different data chunks along different paths in parallel, and finally be re-assembled at the client to reconstruct the whole content.

III. THE DESIGN OF CODA

A. Enhanced Interests Model

From the previous description, the efficient use of the FIB and PIT tables by the Interests for multipath data transmission is a key unsolved problem. For supporting these requirements, we extend the work in NDN [11] where the idea of state forwarding is suggested and fully illustrated.

Detector Interests**:** In detail, we introduce a new type of Interest packet called *Detector* for constructing multipath data transmission. One set of Interest packets flow corresponds to one *Detectors*. For example, Interest packets with the name */prefix/content/01* and */prefix/content/02* from the same set, all map to the *Detector* with the name of */prefix/content:consumer*.

The consumer multicasts the *Detector* packets while sending the normal Interests. Each *Detector* packet is associated with a Time-To-Live (TTL), which is decreased every time the *Detector* is forwarded. *Detectors* do not return the corresponding data packets, even if they reach a cache or a producer that contain a copy of the content. However, *Detector* messages are updating the PIT, by adding an entry for the *Detector* packets. Once a *Detector* packet arrives at a node, it will create an PIT entry with its incoming interface and its name, if this node has not been visited yet by the same *Detector*. As it can be seen from the naming strategy of the *Detector*, the PIT created would match *all* the chunks of data flow with this prefix. Therefore, *Detectors* are sent once at the beginning at the transmission to set up the alternate paths (described in more detail below) for all the packets matching their prefix[1].

B. Multipath Discovery and Transmission

(1)Interests forwarding phase

Initially, the consumers in NDN have no idea about the multiple potential paths for retrieving the requested content. To find out the locations of the copies of the content, several routing algorithms have been devised to build the FIBs in each nodes. We assume here that one such algorithm has populated the FIBs (it is outside of the scope of this paper to investigate such prefix dissemination algorithm).

However, the consumer is not able to divide the content equally to forward the Interests along different paths due to the lack of global information about the producers and content caching nodes. For example, if the consumer has two next neighbors to choose according to its FIB, there is no guarantee that the two possible paths would have similar performance. One may be much slower, and instead of improving the throughput, the one with longer latency would degrade the performance. Considering the trade-off between global information spreading, and efficient multipath construction, we propose the design of a primary path and alternative paths (See the example in Fig. 1).

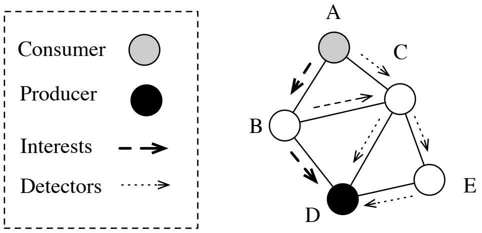


Fig. 1: A simple example of primary path and alternative paths: $path_{ABD}$ forms the primary path while $path_{ACD}$ and $path_{ACED}$ are the alternative paths.

Primary path: With the design of current NDN architecture, the consumer node forwards its Interests according to the NDN forwarding strategy. These Interest packets will then keep being forwarded until they reach the producer (i.e., the nodes who possesses the desired content) or a copy at a caching node. The Interest dissemination in this way finds a path to the producer. We name this path as primary path. By the way, the number of primary path is optional according to the requirements of applications and network policy.

Alternative paths: During this phase, there is no modification to the protocol running with normal Interest packets. For the discovery of alternate paths, the node in the primary path also broadcasts the corresponding *Detector* packets to all the matching FIB interfaces for this prefix when it receives the first packet of the Interests flow. The *Detectors* here are responsible for building the alternative paths for coordinating data transmission on multiple paths. By using the *Detector* packets, we can build the paths for data transmissions in parallel with the forwarding of the regular Interest messages. The information of these paths is stored in the PIT entries of the forwarding nodes.

(2) Content return phase

Instead of returning the content sources information back to the consumer, CODA directly returns the content from the destination of the primary path. Here, there are two major benefits:

- The data transfer starts within one RTT for the notification of content distribution, and still satisfies the principle

---

[1] If supported by the network, *NACK* mechanisms can be used with Interest and *Detectors* to invalidate FIB entries

of content-location decoupling;

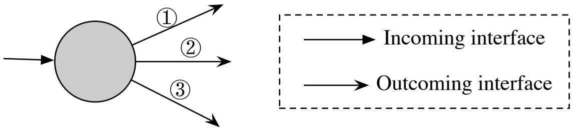
- Less signaling overhead is used for multipath construction. Correspondingly, we have primary nodes in primary path and alternative nodes in alternative paths in CODA.

When a node receives the flow of data packets, it transmits the content by carrying out the following three steps: ① Check the availability of multipath; ② Split the traffic according to the result of step 1; ③ add into the data packet header a path identifier based upon the chosen path, so that the consumer can monitor the per-path performance, and request (respectively, invalidate) a specific path by inserting the path identifier (respectively, excluding that path identifier) in the next Interest message.

(3) Multicast preservation

By selecting a potential alternative path, Data and Interest packets may not follow a symmetric route. Therefore, an Interest may not receive a data response on the primary path. This may create an issue for aggregated Interests in the PIT, especially if the data is required to satisfy more than one requests at this node.

If the response avoids this node on the return path, only one of the clients may eventually receive it. However, it has been shown that Interest aggregation in the PIT has a limited impact on the network performance: less 5% of total Interests on average are subject to aggregation with very small caching budgets [12]. Further, if the node receives Interests for this name prefix, it could still transmit the corresponding content to other consumers by following the primary path (adding the path identifier in the Interest) to ensure that further aggregation is not penalized by the multi-path.



Selected Pending Interest Enties

Type	Lifetime	Name	Requesting interface
Detector	6	/p/n:c	①
Interest	6	/p/n/1	①
Interest	6	/p/n/1	③
Detector	8	/p/n:c	②

Fig. 2: An example of interface classification: from the PIT in the node, *interface-1* could transmit the content in coordination with *interface-2*, and *interface-3* should forward the whole content independently.

For the nodes in the primary path, the data transmission paths are classified into three types according to the corresponding PIT entries, shown in Fig. 2. We should transmit all the data packets to the requesting interface if there is no *Detector* Interest pended. It indicates that one of the consumers wants to retrieve the content via a single path. The effectiveness of multiple paths construction is up to two requirements: there are at least two requesting interfaces of *Detector* available and one of them coincides with normal Interest packets. This is also the major reason accounting for spreading of *Detectors* on the primary path: to reduce the effect on the aggregation mechanism of NDN. With this classification, we can find the interfaces we can use for multipath transmissions.

As for the nodes on the alternative paths, the availability of multipath is much simpler. Since we put the consumer name (or some identifier) on the *Detector* packets, we can use this information to find the requesting interfaces to choose the paths.

C. Traffic Scheduling

Using the entries set up by the *Detectors* in the PIT, a node may have a choice of multiple interfaces for transmission. It needs to know the way to split the traffic into multiple sub-flows transmitted in different paths so that each sub-flow could achieve simultaneous completion in the consumer side. In the design of NDN, the content chunk is the basic unit of data for caching and transmission. Hence, each sub-flow should consist of at least one content chunk in the implementation of a traffic schedule.

(1) Delay Measurement

A key challenge here is to carefully control how many chunks to allocate for each sub-flow. A basic round-robin strategy would work well if all the paths have similar properties. However, this is not a practical assumption. Therefore we propose a delay measurement method to make the path selection decision within a node.

The overall delay is composed of propagation delay, transmission delay, queueing delay, and processing delay. The processing delay is usually negligible compared to other terms in the delay equation. Thus, let W_i be the queueing delay of sub-flow i. The transmission time and processing delay in an intermediate node λ for path i can be represented as service rate μ_λ. The total amount of time spent in the network on path i is:

$$\Delta t_i = W_i + \sum_{\lambda=1}^{k} \frac{S_i}{\mu_\lambda} \tag{1}$$

where S_i describes the packet size of the flow, k is the number of intermediate nodes, and Δt_i is the length of the time between the arrival time of the first packet and the departure time of the last packet. Later, we use μ_i as the average serving rate of sub-flow i. In practice, we try to adjust Δt_i in each paths equally to approximately achieve this principle. Then, for sub-flow i and j, we have:

$$W_i + \frac{S_i}{\mu_i} = W_j + \frac{S_j}{\mu_j} \tag{2}$$

for ensuring the simultaneous completion of each sub-flow.

Suppose the queuing delay is equal of each nodes, thus, the expectation size allocation for each sub-flow can be calculated as follows:

$$S_i \propto \mu_i \tag{3}$$

So far we have shown the policy (equation 3) to schedule packets for traffic splitting. The size of packets outputted at each interface will be determined by the delay and the serving rate of the selected path. Recall that CODA employs *Detectors* to construct multiple transmission paths. The serving rate μ_i of the intermediate nodes can be estimated with the arriving time of *Detectors*. With this information, the traffic splitting mechanism could be implemented in a distributed way.

(2) Sub-flow traffic splitting

In some scenarios, there will be more than one *Detector* for the corresponding data content passing through one interface. It indicates that more paths would be available when transmitting data packets towards this interface. For simultaneous sub-flow completion, we could neither regard this output interface as a single path nor use a combination of delay in each *Detector*. In this paper, we use the hop counter for making an aggregation result for this case. Suppose there are l *Detectors* and the hop count from consumer to the source (content cache or producer) is $\mathcal{H}$. Parameter h represents the distance from the source. Then, we split the traffic according the following equation:

$$S_i \propto \sum_{x=1}^{l} (\frac{\mathcal{H}-h}{\mathcal{H}})^{x-1} \mu_x \tag{4}$$

Equation 4 obeys the following two properties:

- $\frac{\mathcal{H}-h}{\mathcal{H}}$ indicates that if this node is close to the content source (e.g., h is small), there will be more chances in forming efficient multipath.
- We use a power law distribution for managing the possibility of constructing multipath with given l. It means that the *Detectors* whose delay is large would have low contributions in constructing multipath.

We now walk through Algorithm 1, which combines chunk-based transfer, delay measurement and sub-flow traffic splitting. The scheduling algorithm works at the source side. The input consists of the potential network paths and a data buffer that stores the packetized chunk data to be transmitted. The algorithm is invoked whenever a sub-flow can transmit a packet (e.g., has matched entries in PIT). It makes the decision for allocating the coming flow to the next hops. We use line 3 to get the available downstream paths for transmitting the packets with matching *Detector* names. Line $16-21$ handles the sub-flow splitting with the estimated delay measurement in Line $8-15$. The default behavior is to always transmit a set of data chunks with the delay and paths calculated.

D. Data Transmission Control

As discussed above, we construct the multipath for data transmission in a fully distributed way via spreading *Detectors* and constructing the traffic schedule. In order to further support the data transmission with high throughput and reliability, we apply a sliding window data rate control scheme to manage the flow rate of each path. This scheme controls the rate of Interests propagation at the consumers without making any adjustment at any other nodes. The consumer can then choose how to prioritize different paths based upon the observed performance. Another advantage of CODA here is that, the control scheme directly uses the meta message from the data packets, not the Interest packets. The details of the procedure are as follows:

Algorithm 1: Traffic Splitting Algorithm

```
1  INPUT: Content data in the buffer;
2  OUTPUT: data chunks in sub-flow $\tau$;
3  interfaces ← GetMatchedPendingDownstreams($Name$);
4  chunkSet ← GetChunkset();
5  beginNo ← 0;
6  endNo ← 0;
7  $hop\_ratio = \frac{\mathcal{H}-h}{\mathcal{H}}$;
   /* Delay Measuring                                  */
8  $total \leftarrow 0$;
9  for $\tau = 1$ in interfaces do
10     $\mu_\tau \leftarrow 0$;
11     for $x = 1$ in $\tau$ do
12         $\mu_\tau \leftarrow \mu_\tau + (hop\_ratio)^{x-1} \times \mu_x$;
13     end
14     $total \leftarrow total + \mu_\tau$;
15 end
   /* Traffic Splitting                                */
16 for $\tau = 1$ in interfaces do
17     $trans_\tau \leftarrow chunkSet \times \frac{\mu_\tau}{total}$;
18     beginNo ← endNo+1;
19     endNo ← beginNo+ $trans_\tau$;
20     Transmit$_\tau$(beginNo, end No);
21 end
```

Initially, the data packets would be sequenced by its arriving time. We maintain a sliding window w_r for each path to propagate the interest packet in the buffer. When the consumer receive a data chunk from a path, the corresponding size of the sliding window would increase by a/w_r. Here, $a = 1/\sqrt{n}$ and n is the number of paths constructed. Hence, the consumer could choose the path, which have the largest size of the window, as the primary. In particular, the Interests sending rate would equal to the sum of all the windows size, since we have only one primary path. Besides, the sliding window scheme here would also play the role in realizing the congestion control or handle some link failure errors. If some data packets did not arrive in time, the window size will be cut off half of its size (by $w_r/2$), and re-transmit the related Interests to the fastest path, excluding the path that had poor performance.

Since CODA can choose to transmit data via a single path or multiple paths with the help of the *Detector* packets, we select the fair path allocation algorithm [13] displayed above to manage the data rate.

E. Summary

We now elaborate on how to integrate the CODA algorithm into a real system. Fig. 3 plots the system diagram. At the source side, the data chunks coming from the upstream caching or producer nodes are stored in the buffer, and are then split, scheduled, and transmitted by the traffic scheduler. We design a module for measuring and predicting network

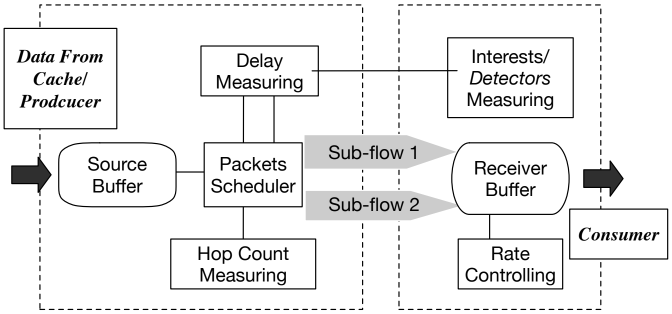


Fig. 3: Coda Diagram

conditions obtained from the *Detectors*. Working with hop count measurement, the delay manager keeps track of the states of the potential multipath constructed with the interface and makes decisions on traffic allocation. The consumer side logic is much simpler. It passively receives the data chunks after spreading the Interests, and reassembles them in the receiver buffer, and delivers the in-order data to the applications. The sliding window based on consumer side is used to further control the rate of Interest forwarding with the feedback of data packets.

IV. PERFORMANCE EVALUATION

A. Simulation Setup

We have implemented our proposals in ndnSIM [14], and made a detailed comparison against some relevant benchmarks. We use the application of producer-consumer to test our proposals. We randomly choose some of the nodes in the tested topology as consumers, and some of them as producers. By default, the expected number of consumers and producers are 16 and 4 respectively. The consumer sends the interests in batch, and the producers respond with the data which is represented by a particular prefix. We use a network topology with 191 nodes and 351 links to evaluate the performance. The diameter of this topology is 20. We obtain similar results with other topologies. Unless otherwise specified, we use Least Recently Used (LRU) as the default caching policy, which is provided by ndnSIM.

We evaluate and compare the performance of the following forwarding strategies, including both single path and multipath methods:

- Flooding: The flooding strategy contributes to improving the delivery reliability via finding as many potential producers as possible. It sends every interest to all interfaces (except downstream).
- Best-route: Normally, this strategy achieves the following goals with global routing calculation: (1) it discovers the paths to the newest content copies; (2) it forwards the Interests for the required content via the best-performing path (typically the shortest path).
- MSRT: MSRT [7] is a Multi-Source Request and Transmission mechanism that uses Probe Interests to obtain the content sources information for the consumers. Despite the overhead in aggregating the sources information to the consumers, it is an ideal multipath data transmission solution for NDN.

B. Network Throughput

(1) Stable Network Conditions

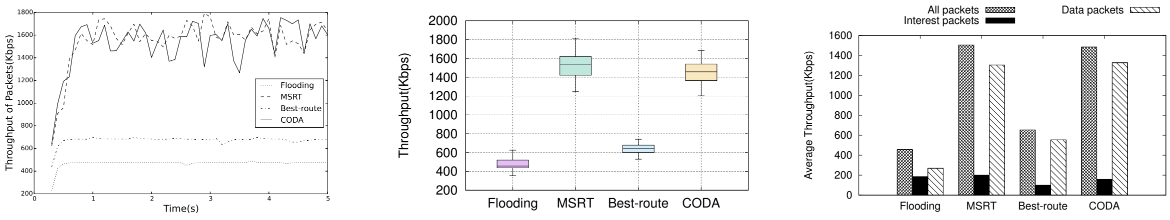
We now evaluate the throughput performance of CODA. Initially, we send $2,000$ Interest packets every second. In this scenario, the content could stay in the cache permanently, and the bandwidth of each link is $1Mbps$. For each strategy, we get the throughput results towards the selected consumer nodes.

Fig. 4(a) demonstrates the achieved throughput of the consumer nodes when running the four different Interests forwarding and data transmission algorithms. We only capture the first 5 seconds since the trends are very similar afterwards. The Flooding method has the worst data rate, which only achieve $490kbps$ in this experiment. Best-route is a bit higher and reaches $700kbps$. From this figure, we can observe that there is a definite improvement when using multiple paths for data transmissions. CODA and MSRT are all around the rate of $1,500kbps$, which achieve more than twice the throughput of Best-route. Fig. 4(b) shows further details of the throughput distribution with the consumers. As listed in this figure, CODA has performance similar to MSRT, which implies that it is able to construct multipath for data transmission. Fig. 4(c) decomposes the throughput into Interest packets and Data packets. As shown in the figure, CODA has a more satisfactory performance in the rate of data packets. In average, MSRT needs an extra $30kbps$ for transmitting Interests compared with CODA. In addition, this simulation is more favorable to MSRT since we neglect the changes of the content in producers or caches when the network is stable.

(2) Varying Network Conditions

We next evaluate how CODA performs under changing network conditions. In this scenario, we consider fluctuating bandwidth and temporary content placement. We take a normalized distribution approach ($\mu = 1, \sigma = 0.5$) to randomly emulate the varying bandwidth. With this settings, content location might change dynamically, causing some retransmissions in MSRT when there is some inconsistent information between the consumers and content sources.

Fig. 5(a) shows the data rate changes with time. A performance decrease could be seen for all the four strategies when compared with the static network. As shown in the figure, the data rate of MSRT and CODA drops from about $1,500kbps$ to $1,350kbps$. Even though, the benefits of multipath transmission hold still as they are able to speed up the throughput by $1000kbps$ compared with Flood and Best-route. A further illustration of the data rate in each strategy is demonstrated by Fig. 5(b). The average throughput of CODA is $1,338kbps$ while MSRT is only $1,298kbps$. It shows that CODA has a better ability to adapt to the changes of the network conditions. Further, Fig. 5(c) displays the ratio of dropped packets. The drop ratio of CODA is around 11% less than MSRT, which is reduced by $2.14\times$ with Best-route, and $5\times$ over Flooding. The data drop ratio is also very low in

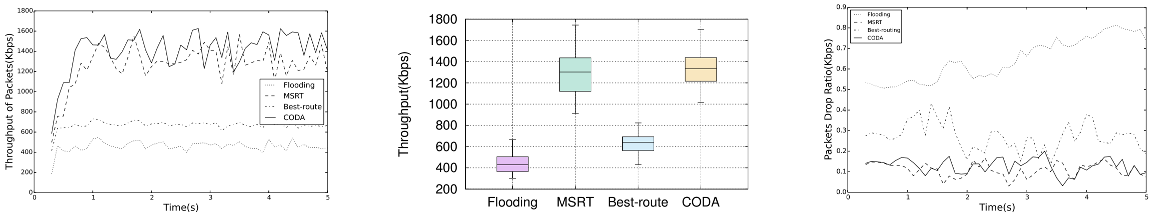


(a) Time

(b) Distribution

(c) Decomposed

Fig. 4: Performance of throughput in stable network



(a) Time

(b) Distribution

(c) Drop rate

Fig. 5: Throughput performance in varying network

MSRT since it could control the data transmission rate in its design. However, the major drawback is that it needs to collect the information of multiple sources periodically while CODA could transmit the data with multiple paths in real time.

C. Data Reduction and Overhead

We present the results for data consumption. In this scenario, each consumer sends $2,000$ interest packets per second for 6 seconds with the stable network settings. Fig. 6 displays the relative number of the packets forwarded in all the network nodes, as well as the breakdown of Interest and Data packets. It is noticeable that the number of Interest packets vary considerably, whereas the transmitted data packets are similar for Best-route, MSRT and CODA. MSRT has the largest number of Interests duplicated in the network, which is about $4\times$ larger when compared with Best-route. MSRT also sends $1.5\times$ more Interest packets than CODA since there is no need for the information feedback in CODA. Flooding induces the most data packets (around $2\times$ with others), as more Interests reach the content copies, resulting in a lot of these data packets dropped on the return path since they correspond to duplicates.

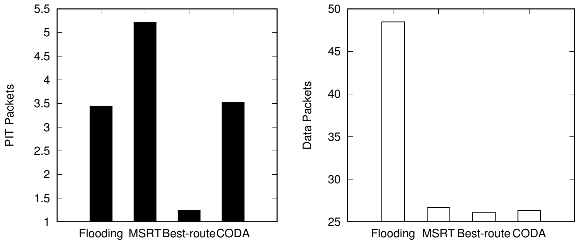


Fig. 6: Overhead in Interest and Data packets

Fig. 7 plots the CDF of the Interest packets throughput among the nodes involved in transmission. Best-route has the lowest overhead in sending Interest packets, and all the nodes receive about $100kbps$ since this method uses a single path to retrieve the content. MSRT requires the most network resource for Interest packets, and the throughput in each nodes varies greatly from the table. Compared to MSRT, CODA uses less overhead in Interest packet, since it only floods *Detectors* once. As the life time of *Detectors* is related to a given value (TTL), CODA has more cost in transmitting Interests packets compared to Flooding.

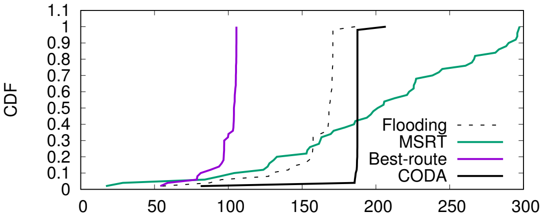


Fig. 7: Distribution of Interests throughput

Since one Data packet is 60 times larger than one Interest packet (in the ndnSIM default implementation), it implies that a few more Interests is an acceptable overhead to improve the performance of NDN.

From the simulations in this scenario, we see that CODA cuts down half of the data redundancy compared with the Flooding method. The overhead due to the extra *Detector* Interests is limited compared with Best-route, and is much smaller than the overhead of MSRT. The results illustrate that CODA is efficient in constructing multipath data transmission with a fully distributed solution.

V. Related Work

Interests forwarding and data transmission in NDN have attracted a lot of scientific research. In this section, we will give a brief survey of the related work from this perspective.

INFORM [15] uses an iterative algorithm to make use of availability of content in all caches and retrieves the content from nearby caches. On-demand Multi-Path Interest Forwarding [6] allocates traffic to disjoint paths via a weighted round-robin scheme. Probability-based Adaptive Forwarding [16] is a solution inspired by ant colony optimization, which selects the forwarding interfaces based on an RTT distribution. Stochastic Adaptive Forwarding [5] imitates a self-adjusting water pipe system, guiding and distributing Interests through network crossings circumventing link failures and bottlenecks. It employs overpressure valves enabling congested nodes to lower pressure autonomously. Carofiglio et al. [17] proposed an optimal forwarding strategy (RFA) via solving the multi-flow minimum-cost problem approximately, which uses the number of Pending Interests as the approximate metric to balance the congestion on different forwarding interfaces. In PTP [8], the consumers specify the transmission path of each Interest packet via a tag (i.e. path label), and the routers follow the tag's instruction to forward the Interest packet to the specified interface. In [18], a network coding based approach is introduced for supporting multipath transmission in ICN. As for MSRT [7], there are additional costs for waiting for the information feedback to the consumers in forming multiple transmission paths. DIVER [19] also extends the Interest model, but mainly focuses on reducing data redundancy to improve the throughput, without taking the capability of multipath transmission of NDN into account.

VI. Conclusion

We have presented the design and evaluation of CODA, a solution that significantly enhances content delivery in NDN by fulfilling multipath data transmission in a distributed and cooperative manner. To achieve this, a new Interest packet type, namely *Detector*, is introduced to build multiple paths for transmitting the corresponding data packets in parallel with a traffic splitting mechanism. Our experimental results show that CODA uses fewer Interests to build multiple data transmission paths compared with MSRT. Moreover, CODA can achieve a throughput twice as high as the best single path routing method, and has the highest average transmission rate under various network conditions. In addition, only minor modifications of NDN are needed for CODA to support cooperative multipath data transmission — existing NDN routers can support CODA by enabling the *Detectors* interests forwarding. In the end, although CODA is specially designed for NDN, its ideas can also be ported to other types of ICN network.

Acknowledgment

This work is sponsored by Natural Science Foundation of China (61572280, 61672312, 61433008).

References

[1] V. Cisco, "Cisco visual networking index: Forecast and methodology 2016–2021.(2017)," 2017.

[2] G. Carofiglio, G. Morabito, L. Muscariello, I. Solis, and M. Varvello, "From content delivery today to information centric networking," *Computer Networks*, vol. 57, no. 16, pp. 3116–3127, 2013.

[3] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.

[4] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang *et al.*, "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[5] 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.

[6] A. Udugama, X. Zhang, K. Kuladinithi, and C. Goerg, "An on-demand multi-path interest forwarding strategy for content retrievals in ccn," in *Network Operations and Management Symposium (NOMS), 2014 IEEE*. IEEE, 2014, pp. 1–6.

[7] Q. Li, B. Gan, G. Hu, Y. Jiang, Q. Liao, and M. Xu, "Msrt: Multi-source request and transmission in content-centric networks," in *Quality of Service (IWQoS), 2016 IEEE/ACM 24th International Symposium on*. IEEE, 2016, pp. 1–10.

[8] Y. Ye, B. Lee, R. Flynn, N. Murray, and Y. Qiao, "Ptp: Path-specified transport protocol for concurrent multipath transmission in named data networks," *arXiv preprint arXiv:1802.02828*, 2018.

[9] C. Tsilopoulos, G. Xylomenos, and Y. Thomas, "Reducing forwarding state in content-centric networks with semi-stateless forwarding," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 2067–2075.

[10] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Modeling data transfer in content-centric networking," in *Proceedings of the 23rd international teletraffic congress*. International Teletraffic Congress, 2011, pp. 111–118.

[11] C. Yi, J. Abraham, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "On the role of routing in named data networking," in *Proceedings of the 1st international conference on Information-centric networking*. ACM, 2014, pp. 27–36.

[12] A. Dabirmoghaddam, M. Dehghan, and J. Garcia-Luna-Aceves, "Characterizing interest aggregation in content-centric networks," in *IFIP Networking Conference (IFIP Networking) and Workshops, 2016*. IEEE, 2016, pp. 449–457.

[13] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, "Design, implementation and evaluation of congestion control for multipath tcp." in *NSDI*, vol. 11, 2011, pp. 8–8.

[14] A. Afanasyev, I. Moiseenko, L. Zhang *et al.*, "ndnsim: Ndn simulator for ns-3," *University of California, Los Angeles, Tech. Rep*, vol. 4, 2012.

[15] R. Chiocchetti, D. Perino, G. Carofiglio, D. Rossi, and G. Rossini, "Inform: a dynamic interest forwarding mechanism for information centric networking," in *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*. ACM, 2013, pp. 9–14.

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

[17] G. Carofiglio, M. Gallo, and L. Muscariello, "Optimal multipath congestion control and request forwarding in information-centric networks: Protocol design and experimentation," *Computer Networks*, vol. 110, pp. 104–117, 2016.

[18] M.-J. Montpetit, C. Westphal, and D. Trossen, "Network coding meets information-centric networking: an architectural case for information dispersion through native network coding," in *Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications*. ACM, 2012, pp. 31–36.

[19] I. V. Bastos and I. M. Moraes, "Diver: A diversity-based search-and-routing approach for named-data networking," in *Global Communications Conference (GLOBECOM), 2016 IEEE*. IEEE, 2016, pp. 1–7.