Flow-based NDN Architecture

Xiaobin Tan*†, Zhifan Zhao*†, Yujiao Cheng* and Junxiang Su*†

*Laboratory for Future Networks, University of Science and Technology of China, Hefei, China, 230027
†CAS Key Laboratory of Technology in Geo-spatial Information Processing and Application System, University of Science and Technology of China, Hefei, China, 230027
Email: xbtan@ustc.edu.cn, {zzfan@, cyj1993@, jxsu@}mail.ustc.edu.cn

***Abstract*—Named Data Networking (NDN) architecture promises significant advantages over current Internet architecture by replacing its host-centric design with a content-centric one. In NDN, the mode that one Interest packet gets one Data packet can quite easily lead to Interest flooding and a huge number of the related entries. Moreover, the absence of predefined connections is a challenge for NDN to efficiently manage the successive requests from consumers or the in-network concurrent requests. In this paper, we argue that it is necessary for NDN to support flow transmission mechanism aimed at improving transmission performance, and based on it we design flow-based NDN (*f*-NDN) architecture which can not only achieve overload decreasing by packing successive Interest packets but also performance improvement and load-balance by multi-path. Simulation of our architecture is carried out in different network scenarios to evaluate the performance of our architecture. Evaluation results show that the proposed architecture operates better than NDN architecture in many aspects such as transmission efficiency and system load including reducing the number of Interest packets and lookup operations performed on the related tables in routers.**

I. Introduction

The essence of Named Data Network (NDN) [1] architecture is that the network layer provides users with contents instead of communication pipelines between hosts. To obtain data, the requester should send out an Interest packet, which carries the name of the desired data (*e.g. /supplied-name/version/chunk-number*). Each router has a *Content Store (CS)* to make a cache of some content and maintains a *Pending Interest Table (PIT)* to keep track of the currently unsatisfied Interest packets. Arriving Interest packets are forwarded using name-based forwarding by looking up the data name in the locally maintained *Forwarding Information Base (FIB)* when there are no pending Interest packets with the same name in PIT and no corresponding entry in CS. Actually, in NDN, not only the receiver, but also in-network nodes may perform dynamic packet-by-packet request scheduling in a purely dynamic fashion by taking on-the-fly decisions on forwarding interfaces.

The mechanism of one Interest packet getting one Data packet in NDN may be more flexible when dealing with the problem of mobility, path failure, and multi-path forwarding etc. However, this may cause overload problems due to the countless Interest packets. Considering a request for a large-size content object (e.g. a 1GB video) in NDNx prototype [2] (the maximum size of the data packet is 8800 Bytes), 1GB video will be split into approximately 120,000 chunks. Thus, the same number of Interest packets will be generated to get the corresponding Data packets just for a 1GB video. Furthermore, each Interest is forwarded using name-based forwarding, with each content router resolving the content name to an outgoing interface by involving multiple accesses to the off-chip high-latency memory to facilitate the LPM operation. In the case of millions of such requests, a noticeable increase in the cost of performing lookups on three tables of NDN may be observed.

Although there are many researches that operate on the per-packet basis to reduce the number of lookups on the forwarding tables or to decrease the size of the PIT/FIB (e.g. [9] [12] etc.), flow-based method still have great advantages when it comes to improving traffic efficiency and reducing network overload. For example, in [6], the authors present a method that exploits the correlations in user traffic to create active flow states in content routers to bypass the default forwarding for future requests, just like IP network with the features of creating network pipes between hosts identified by addresses, which can address the recurring lookups on FIB overhead problem. And in [5], they introduce a virtual control plane to optimize the flow transmission in actual plane which can maximize the user-demand. However, these approaches may offer limited gains especially when the *multi-path transmission* occurs, and a comprehensive analysis of the joint problem of flow transmission and multi-path transmission in NDN still needs to be concerned.

In this paper, based on the core idea of *Packing Interests*: the continuous requests for the successive chunks of the same content object can be merged into one flow Interest packet, we propose flow-based NDN (*f*-NDN) architecture. The proposed architecture concerns the flow transmission in a connectionless and stateful transport model. The main contributions of this paper are as follows:

- We propose a new Interest packet format that takes advantage of the pull model to acquire contents. And it can dramatically decrease the number of Interest packets. To some degree, we solve the problem of Interest flooding.
- Our proposal can decrease the number of lookups on the related forwarding tables without causing more system load by the method of flow transmission.
- *f*-NDN can achieve multi-path transmission, which significantly improves service reliability and network flexi-

bility.

- We propose a forwarding strategy on a per-flow basis, and it is implemented on the ndnSIM prototype to present the performance of our f-NDN architecture.

This paper is organized as follows. The NDN architecture's problems and challenges are summarized in Section II. The architecture's design details including architecture description, working mechanism, and forwarding protocols are presented in Section III. Section IV is the performance evaluation of our proposed architecture. Finally, we concludes our paper in Section V.

II. Problem Analysis

NDN transport model brings new challenges that TCP/IP based network, even in its multi-path versions, can not address, and it motivates the definition of novel transport control and forwarding mechanisms. However, in TCP/IP based network, in regard of its connection-based character, the multi-path solutions (e.g. [13]) perform load-balancing mechanisms over static paths which are precomputed by a routing protocol. Instead, in NDN, it's a great challenge to achieve more efficient flow transmission and multi-path transmission due to the absence of predefined connections and the lack of knowledge of available source(s).

Previous efforts have always been devoted to exploiting the transfer capacity of multiple paths on the per-packet basis [7] [10]. Actually, if we want to better exploit the transfer capacity of network, the processing capacity of in-network nodes and the mechanism of multi-path transmission should be taken into account. Cisco Visual Networking Index [11] suggests that the sum of all forms of video traffic is expected to reach 80-90% of the global consumer traffic by 2019. Every second, nearly a million minutes of video content cross the network. In NDN, the mode of one Interest packet getting one Data packet can quite easily lead to Interest flooding and a huge number of the related entries. Reducing the number of interests can make a significant improvement in the network performance by decreasing the overload of the whole network. In this paper, the proposed architecture focus on two aspects.

1) Flow transmission: Flow Interest mechanism can not only reduce the number of Interests, but also organize the flow requests in a more efficient way. It has especially better performance in terms of aggregation to reduce the repetitive transmissions for multiple requests for the same data. For example, different flow Interests which carry the flow request for the different successive chunks of the same content object can be aggregated into one flow Interest. Furthermore, different flows can enjoy the different services according to applications' requirement and network condition. Indeed, the operation on the flow Interest greatly simplifies the problem of flow control. But the flexibility brought by the absence of predefined connections is also a challenge for the flow transmission.

2) Multi-path transmission: As for the multi-path transmission in NDN, it means users can continuously request for the desired content object from multi-sources but not at the same time. And the Interest forwarding is controlled by in-network nodes which take on-the-fly decisions on forwarding interfaces for each arriving Interest. Furthermore, interface ranking is according to a three colors scheme [8] (implemented in the NDNx Prototype [2]) which makes no discrimination of different Interests. Instead, users in f-NDN can merge the continuous requests for the successive chunks of the same content object into one flow Interest and one flow Interest can also be split into many sub-flow Interests, which means the flow Interest can be satisfied by more than one repository at the same time.

Research result on multi-path routing in IP has demonstrated that flow throughput and network traffic capacity are greater when congestion control is performed in a coordinated manner over the multiple paths [4]. In NDN, the same problem has also been discussed in [7], but it needs to acquire the available paths between the end-user and the existent caches to determine fast and slow routes by explicit labeling. Actually, our goal is to support Multi-path transmission fundamentally, and it fully utilizes the flexibility brought by the absence of predefined flow transmitting.

III. Proposed Architecture

The goal of this section is to present how the f-NDN architecture is designed with flow and multi-path transmission taken into account.

A. Architecture Description

In order to support the flow transmission, we define the format of flow Interest by including some additional fields to primary Interest packet head. Moreover, we make some changes to the PIT/FIB to make the forwarding of the flow Interest more efficient.

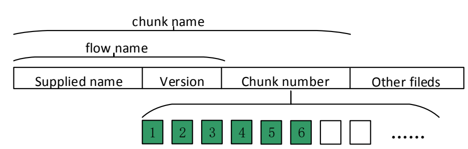

Fig. 1: *Flow name*

1) Flow Name: A summary representation of the hierarchical name structure envisaged in f-NDN is depicted in Fig. 1. The chunk name can uniquely clarify the Data packet and the flow name distinguishes the object with the supplied name and version.

NDN allows name aggregation, e.g., */ustc/video* could correspond to an autonomous system originating the video and each chunk of the data split will be added into that name. In this way, we can regard the "*flow name*" as the name of combination of total chunks. For example, */ustc/video/demo.mpg* is corresponding to the first chunk to the end chunk of the *demo.mpg* (the prefix: */ustc/video*).

2) Flow Interest: To support "*packing Interests*", we have a new design of the Interest packet called flow Interest. We include three additional fields in the primary Interest packet header: *Flow Type* (f-Type) field, *Start Chunk of flow interest* (SC), and *End Chunk of flow interest* (EC) field.

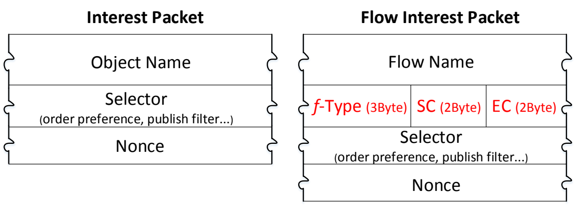

Fig. 2: Interest packet and flow Interest packet

In flow Interest packet (Fig. 2), the 3-Byte field of *f-Type* have strong flexibility and expansibility which can meet the different service requirements. For example, in our implementation, the basic functions are representing the type of this Interest (1-bit), determining whether this flow Interest can be split (1-bit), and the limiting condition of the flow Interest split (e.g. link condition and the maximum of sub-flows). Additionally, the rest bits can also be used to other service functions, such as Quality Of Service (QOS), priority support, and rate control etc. The *SC* and *EC* fields represent the start and the end chunk of the request field[1].

3) Flow PIT (f-PIT): To maximize the utility of the PIT, PIT entries need to be timed out pretty quickly, somewhere around packet round-trip time. On the contrary, if they are timed out prematurely, Data packet will be dropped, and the consumer will have to retransmit Interests. Actually, a flow Interest carries more than one request for the content chunks, which means the PIT should meanwhile generate the same number of PIT entries to record the different requests. However, corresponding Data packets do not arrive at the same time and some of the later reached Data packets will be dropped.

In f-NDN, we organize the PIT entries in a "flow" way and the structure is $\{$*Prefix, (Start Chunk_1, End Chunk_1, Face IDs, Timer_1), ..., (Start Chunk_n, End Chunk_n, Face IDs, Timer_n)*$\}$. Considering the aggregation of the same flow request, f-PIT needs three operations: merge, update, and split.

When a flow Interest arrives (Fig. 3), if there exits a matched f-PIT entry (the request field of the same content is included in the corresponding f-PIT entry) and the incoming face is already in the pending interface set, then this flow request will be merged into the corresponding entries by simply resetting the lifetime of this f-PIT entry. Otherwise if the incoming face is not in the pending interest set, then the new face will be added into the entry's face ID field when the request field is exactly the same as the matched entry, or it will be recorded in a new entry when the request field of the interest is included but not the same as the matched entry.

[1]In this paper, we generally think the request in the flow Interest is the successive chunks of the same content object.

Moreover, if only a part of the flow request matches a f-PIT entry, the matched part is processed using the mechanism mentioned above and the rest part is recorded in a new entry.

When a Data packet arrives, if the coming Data packet is the start(end) chunk of the matched f-PIT, the corresponding f-PIT entry should update the Start(End) Chunk field and Timer. However, if the coming Data packet is the middle chunk of the matched f-PIT entry, the corresponding f-PIT entry will split into two.

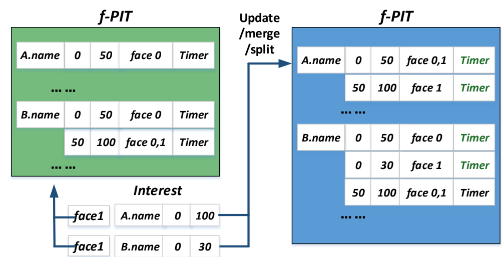

Fig. 3: The structure of f-PIT

4) Flow FIB (f-FIB): The forwarding of the flow interest is just like a flow control problem which determines the total face rate, and also like a routing forwarding problem which decides how to split the total flow interest among a set of least congested faces. In-network nodes perform dynamic flow-by-flow packet request scheduling by taking on-the-fly decisions on the selection of forwarding interfaces. It is necessary to update the interfaces' weight parameters due to the everchanging network conditions.

To make the forwarding of the flow Interest more efficient, we add a new field to the original FIB entry that contains state information of all output interfaces for each flow. The general structure for f-FIB is $\{$*Prefix, (Face_1, Value), ..., (Face_n, Value)*$\}$, which is kept on all output interfaces and is real-time updating. When the matched f-FIB entries have multiple available output interfaces for a flow Interest, the router can acquire the price of each interface by checking the Value field to split the flow Interest or select the minimum cost interface.

B. Working Process

Next, we will present the forwarding process of flow Interest in f-NDN.

1) f-FIB State Update: To manage the forwarding of the flow Interest, we should set the value of the interface for each arriving flow (content object).

Let us consider the download of a given content s, and in-network node i has available out interfaces $J = \{1, 2, ..., n\}$. Let c_j be the capacity of link l for each interface and $c = [c_1, c_2, ..., c_j]$. Let $x^j(t)$ be the sum of transmission rates on the interface j at time t.

First, we give the current price of each interface of node i and define $p^i_{s,j}(t)$ as the interface prices of node i at time t. The interface price algorithm can be modified to

$$p^i_{s,j}(t+1) = [p^i_{s,j}(t) + \gamma(x^j(t) - c_j)]^+,$$

then we can deduce the aggregate price on each interface from observed round trip time (RTT) by using a step size β/c_j:

$$p^i_{s,j}(t+1) = [p^i_{s,j}(t) + \frac{\beta}{c_j}(x^j(t) - c_j)]^+, \qquad (1)$$

The number of pending Interests reflects (i) the path length and response time associated to a given content; (ii) the quality of residual path towards repository/closet copy. Actually, the PIT queuing length $b_j(t)$ evolves according to

$$b_j(t+1) = [b_j(t) + (x^j(t) - c_l)]^+,$$

multiplying both sides by β/c_j, we have

$$\frac{\beta}{c_j}b_j(t+1) = [\frac{\beta}{c_j}b_j(t) + \frac{\beta}{c_j}(x^j(t) - c_j)]^+. \qquad (2)$$

Comparing (1) with (2), we see that link price at time t is proportional to the current delay $q_j(t) := b_j(t)/c_j$ at interface j:

$$p^i_{s,j}(t+1) = \beta\frac{b_j(t)}{c_j} = \beta q_j(t).$$

Given end-to-end propagation delay $\delta^i_{s,j}(t)$, hence the aggregate price can be deduced from $RTT^i_{s,j}(t)$:

$$p^i_{s,j}(t+1) = \beta(RTT^i_{s,j}(t) - \delta^i_{s,j}(t)).$$

And with $P_j \equiv RTT^i_{s,j}(t) - \delta^i_{s,j}(t)$, we define $p^i_{s,j}(t+1) = \beta P_j$.

The measure of $\delta^i_{s,j}(t)$ may be very difficult, but it can be estimated by the minimum $RTT^i_{s,j}(t)$ observed from the history samples.

2) *Flow Interest Splitting:* Considering a traffic forwarding strategy in [8], all the Interfaces in a FIB entry are ranked in order to help choose which interface to use. We can define the interface ranking by setting two boundary values (p^1_i, p^2_i) for the price of each face:

- Green ($0 < p^i_{s,j} < p^1_i$): the interface is in good condition.
- Yellow ($p^1_i < p^i_{s,j} < p^2_i$): the interface is overloaded.
- Red ($p^2_i < p^i_{s,j}$): the interface does not work.

Actually, a flow Interest wishes to meet the f-FIB with the Green interfaces, then the in-network node only selects the interface with the minimum $p^i_{s,j}$ to forward the flow Interest. However, the Green ranking interface does not always exist in in-network nodes and this method is difficult to reach load balancing. Overload occurs when the demand exceeds link capacity leading therefore to unacceptable performance for some flows. In this case, single interface has difficulties to perfectly satisfy the need of the flow Interests. Even if we still choose the interface with the minimum $p^i_{s,j}$ to forward the flow Interests as the method proposed in [7] on the per-packet basis, network may not reach the expected performance.

To achieve certain traffic engineering objectives, the network management system needs to balance traffic between multiple paths. For example, sending 40% of traffic on one path and 60% on another could lead to less congestion in the network and it can be formulated as a simple convex optimization problem (assuming that the flow Interest carries the request for N chunks of content s and there are λ available interfaces in node i):

$$\begin{cases} \min\limits_{\{n_1,n_2,\ldots,n_\lambda\}} \max\{n_1 p^i_{s,1}, n_2 p^i_{s,2}, ..., n_\lambda p^i_{s,\lambda}\} \\ n_1 + n_2 + ... + n_\lambda = N \end{cases} \qquad (3)$$

Obviously, when $n_1 p^i_{s,1} = n_2 p^i_{s,2} = ... = n_\lambda p^i_{s,\lambda}$, we can easily get the optimal solution values $\{n^*_1, n^*_2, ..., n^*_\lambda\}$. In this way, we can split the flow Interest into several sub-flow Interests, i.e., split the flow request into multiple interfaces according to the price of each interface (Fig. 4).

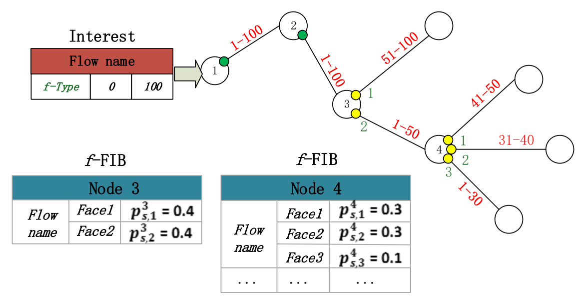

Fig. 4: Flow Interest Splitting

3) *Forwarding Strategy:* The flow Interest forwarding details are shown in Fig. 5. When a router receives Interests for the same name from multiple downstream nodes, it forwards only the first one upstream toward the data producers. And not only the aggregation of different flows can bring up a significant promotion, but also splitting the request of flow Interest into all the available interfaces according to the congestion states of each link is beneficial.

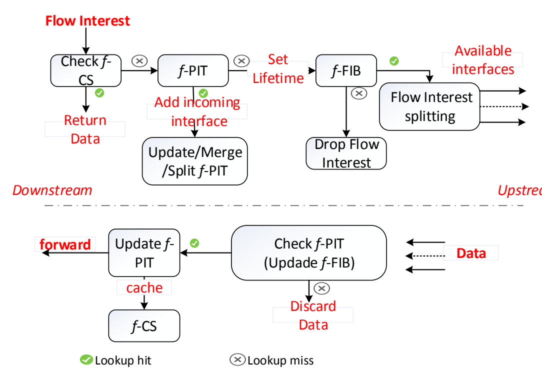

Fig. 5: Forwarding Process at a f-NDN Node

In f-NDN, the price p of output interface is real-time updated on each node. Upon the arrival of each Data packet, the price of a particular output interface is updated. And at the f-PIT timeout, the particular interface will be punished by multiplying the maximum RTT_{max} (from history samples) with 2. At each flow Interest's reception, we check the f-FIB to get the available interfaces, and split the flow request according to the price of each interface (In our implementation, we only split the flow Interest into two available interfaces with the minimum costs). As for the order, the first part of the flow request is expressed into the minimum cost interface, the second part of the flow request expressed into the most-near-minimum cost interfaces (the details are described in Algorithm 1). And, specially, flow Interest may have particular settings in its *f-Type* field. For example, some flow Interests may have set threshold value, which means in-network nodes can only split the flow Interest when there are more than one available interfaces with the price under the threshold value. Moreover, in order to obtain more stable content stream, some end-users can forbid any splitting operations for their flow Interests.

Algorithm 1 Interest Dividing

```
At Interest_Packet_Reception (set, flow_name, face_id)
if (CACHE_miss && PIT_miss) then
    Init_get(Offset, End) from set
    Init_cal(face_list, price_list) from f-FIB
    Get the N = {n*_1, n*_2, ..., n*_λ}
    for all i < λ do
        Interest.init_get(name, SC, EC)
        Transmit(Interest, PIT.Match(Set).Face[i])
        Offset ⇐ Offset + n*_i
    end for
end if
```

IV. PERFORMANCE EVALUATION

In this section, we focus on the evaluating performance of the load balance brought by multi-path transmission and the transmission efficiency improvement including overload decreasing and the better reactions towards the congestion situation.

To show the performance of multi-path transmission in f-NDN, we implement the protocols described above in NS-3 based Named Data Networking (NDN) simulator (ndnSIM) [3] by developing additional modules and changing some structures of the ndnSIM prototype. The detailed settings are showed in TABLE I. We employ a simple network scenario with a good network condition (see Fig. 7), in which users are connected to two repositories via two paths and the capacity of Data packet generation can be adjusted in repo1 and repo2. We observe the load change of $link_{3,4}$ and $link_{3,5}$ when we adjust the produce capacity of repo1 and repo2 (the unsatisfied flow Interests will be resent soon). Evaluation results are showed in Fig. 6.

TABLE I: Simulation Parameters

Simulation parameters	Value
Producer Capacity (per packet)	0.0001s
Interest Packet Size	35B
Delay Time	10ms
Queue(Maxpackets on the link)	20
CPU	2.60GHz
Memory	4G

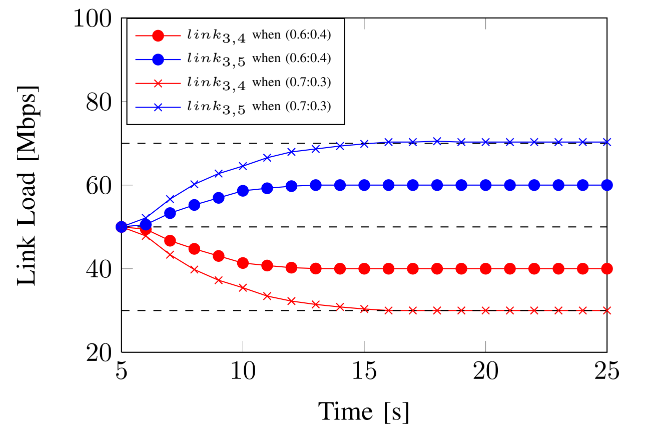

Fig. 6: The change of the link load of $link_{3,4}$ and $link_{3,5}$ when we adjust the produce capacity of repo1 and repo2 from 0.1:0.1(ms/per-packet) to 0.6:0.4(ms/per-packet) and 0.7:0.3(ms/per-packet).

A. Overload Considerations

In this scenario, we focus on the overloaded traffic situation including packet processing, tables maintaining, and the number of Interests. Notice that the total time (τ_{total}) includes: the forwarding time of Interest packet (τ_{ip}) and Data Packet (τ_{dp}), the lookups time on the CS/PIT/FIB (τ_l), the preset response time of producers ($\tau_r = 10ms$). That is,

$$\tau_{total} = \tau_{ip} + \tau_{dp} + \tau_l + \tau_r$$

The setting rate field in flow Interest is equal to the generation rate of Interest packet in NDN, which means the τ_{dp} is equal to the τ_{f-dp}. We have set both response time of producers $\tau_r = 10ms$. Actually the time of dealing with the flow Interest is more than original Interest packet, but the main impact factor is the number of Interest packets. To completely consider all factors, we can approximate the overhead on packet processing by measuring the total processing time.

Scenario 2 topology is shown in Fig. 8. there are four consumers to request the same content object from repo1 and repo2. The four consumers start to request the content respectively at 0s, 5s, 10s, 15s. Especially, the network nodes in f-NDN is fixed to select the best two interfaces for the flow Interest, and the forwarding strategy in NDN is also the same for the Interest flows. Each node(#3,#4,#5) records the time of dealing with the Interest packet. TABLE II demonstrates the total number of Interests sent to acquire the whole content in network and the overall time for the nodes (#3,#4,#5) to deal with the passing Interests. Obviously the improvement of reducing the overhead on packets processing is remarkable.

TABLE II: Interest Processing Time

Node ID	f-NDN / NDN		
	Number of Interests	Per-packet processing time [s]	Total time [s]
3	239 / 1265	0.0124 / 0.092	3.038 / 11.638
4	318 / 1465	0.0124 / 0.092	3.943 / 13.478
5	237 / 1270	0.0124 / 0.092	2.939 / 11.684

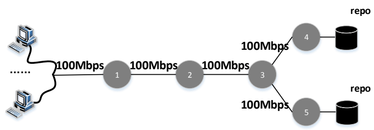

Fig. 7: Scenario 1 topology

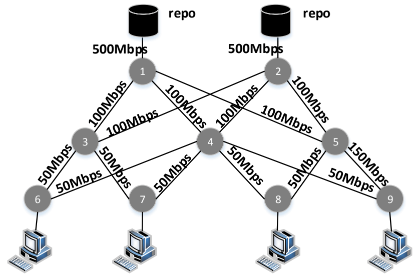

Fig. 8: Scenario 2 topology

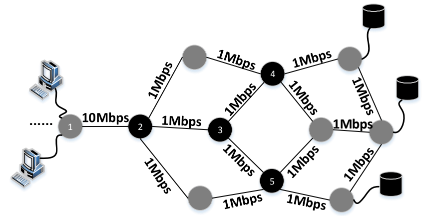

Fig. 9: Scenario 3 topology

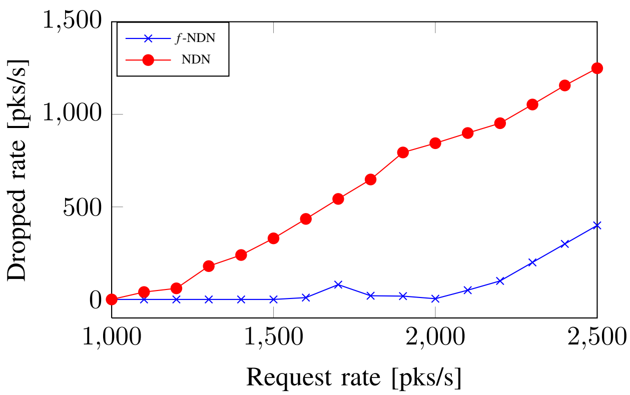

Fig. 10: The number of dropped Data packet per second (pks/s) in different request rate.

B. Congestion Considerations

In a forwarding architecture, the most critical performance measure is the *forwarding capacity*, and whether or not the proposed solution can support the targeted rates. In this scenario, we consider a typical network scenario to evaluate the performance of Multi-path transmitting when the network is congested, which has also been discussed in Sec. III-C. The topology is shown in Fig. 9, the bandwidth of access link is 10Mbps, other links 1Mbps. In this case, the network can easily become congested when increasing the user's request rate. The associated FIB entries are set to make all the output interfaces available at nodes (#2,#3,#4,#5). To represent the performance of congestion control and forwarding protocols adapted to different network conditions, we record the dropped Data packets with increasing request rate (Fig. 10). The rise of Dropped Data packet in NDN occurs immediately when each of the link can not satisfy the arriving Interests. In *f*-NDN, the node can select more than one available interface to meet the sudden growth of demand due to the advantage of the flow Interest.

V. Conclusion

Flow-based NDN architecture is a new architecture whose operations can be derived from current practice. Its design reflects our understanding of the strengths and limitations of the current NDN architecture. In this paper, we present a detailed analysis of the proposed architecture and investigate its robustness during path congestion. We show the significant improvements of performance achieved in forwarding capacity and energy utilization when in the mode of flow transmission and multi-path transmission. However, there still exits many works to do in this architecture. For instance, the rate control may be a challenge in *f*-NDN, although users can manage the flow rate by setting the *f-Type* field or adjusting the size of the flow Interest according to their needs and network conditions.

VI. Acknowledgments

This work was supported by the State Key Program of the National Natural Science Foundation of China under Grant 61233003, and National "863" Project of China under Grant 2014AA06A503.

References

[1] Zhang L, Estrin D, Burke J, et al. Named data networking (ndn) project. *Relatrio Tcnico NDN-0001, Xerox Palo Alto Research Center-PARC*, 2010.

[2] NDNx. http://named-data.net.

[3] NS-3-based NDN simulator. http://ndnsim.net.

[4] Key P, Massouli L. Fluid models of integrated traffic and multi-path routing. *Queueing Systems*, 53(1): 85-98, 2006.

[5] Edmund Yeh, Tracey Ho, Ying Cui, and Michael Burd. VIP: Joint traffic engineering and caching in Named Data Networks. In *Computing, Networking and Communications (ICN)*, 2014.

[6] Azgin A, Ravindran R, Wang G. Flash-forward CCN: flow-driven forwarding architecture for content centric networks. Proceedings of the 1st international conference on Information-centric networking. ACM, 2014: 189-190.

[7] G. Carofiglio, M. Gallo, L. Muscariello, M. Papalini, and Sen Wang. Optimal Multi-path Congestion Control and Request Forwarding in Information-Centric Networks. In *proc. of IEEE ICNP*, 2013.

[8] Yi C, Afanasyev A, Wang L, et al. Adaptive forwarding in named data networking. *ACM SIGCOMM computer communication review*, 42(3): 62-67, 2012.

[9] Haowei Yuan and P. Crowley. Scalable Pending Interest Table Design: From Principles to Practice. IEEE INFOCOM, 2014.

[10] Udugama A, Goerg C. Adaptation of multi-path forwarding strategies for ccn to operate under mobility. *CCNxCon2013*. Parc, 2013.

[11] Cisco. Cisco Visual Networking Index: Forecast and Methodology, 2014-2019. tech. rep., 2014.

[12] Fukushima, Makoto, Atsushi Tagami, and Toru Hasegawa. Efficiently looking up non-aggregatable name prefixes by reducing prefix seeking. *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*. IEEE, 2013.

[13] Wischik, Damon, et al. "Design, Implementation and Evaluation of Congestion Control for Multipath TCP." NSDI. Vol. 11. 2011.