



An analysis of NDN Congestion Control challenges

Adrien Thibaud, Julien Fasson, Fabrice Arnal, Renaud Sallantin, Emmanuel Dubois, Emmanuel Chaput

► **To cite this version:**

Adrien Thibaud, Julien Fasson, Fabrice Arnal, Renaud Sallantin, Emmanuel Dubois, et al.. An analysis of NDN Congestion Control challenges. 2019. hal-02314169

HAL Id: hal-02314169

<https://hal.archives-ouvertes.fr/hal-02314169>

Submitted on 11 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An analysis of NDN Congestion Control challenges

Adrien Thibaud^{*†}, Julien Fasson^{*†}, Fabrice Arnal[‡], Renaud Sallantin[‡], Emmanuel Dubois[§] and Emmanuel Chaput^{*†}

^{*}TéSA Laboratory, firstname.name@tesa.prd.fr

[†]ENSEEIH/IRIT/University of Toulouse, firstname.name@enseeiht.fr

[‡]Thales Alenia Space, firstname.name@thalesaleniaspace.com

[§]Centre National d'Études Spatiales, emmanuel.dubois@cnes.fr

Abstract—Named Data Networking (NDN) proposes to change the core of the Internet. Based on mechanisms successfully used in P2P or CDN, it focuses on content and thus the Quality of Experience of users. Such an ambitious plan raises great challenges: caching, multipath, multi-producers, multi-consumers and security. This paper focuses on one of them: the control of congestion. Several studies have proposed congestion control solutions that fall into three kinds: the end-to-end solution, the hop-by-hop type and the hybrid one. However, the community lacks proper evaluations of such specific algorithms. In this work, we have implemented representative solutions on ndnSIM. In a first step, we have tested them on a small scale topology to ease their performance analysis and highlight their strengths and weaknesses. We complete this study with simulations on larger networks in order to confirm our conclusions. Furthermore, all results are reproducible. Eventually, the paper drives a discussion on how application needs could be considered in the design of a NDN congestion control.

Index Terms—Congestion Control, ICN, NDN

I. INTRODUCTION

Nowadays, the content retrieving is one of the main use of the Internet. The IP stack alone does not suit well this service where high Quality of Experience (QoE) is expected rather than pure best effort delivery. This is why overlay networks have been designed to provide a better experience to end-users like Content Delivery Networks (CDN), which place popular content in caches closer to their consumers, or Peer-to-Peer networks (P2P) which retrieve data from multiple sources.

The Information Centric Networks (ICN) define a new type of network layer that is data-oriented instead of location-oriented. Named Data Networking (NDN) [1] is an architectural realization of the broad ICN vision with an active research community. NDN uses the mechanisms that forged the success of HTTP, CDN and P2P directly as a basis for the network layer. Its key features are a receiver-driven architecture, caching capabilities of nodes, use of multi-sources and multipath, and native security. The challenge of this architecture is to design a network more compliant with user needs through a content-oriented approach.

However, for each network functions in a NDN network the following questions are asked : "To what extent the use of a name-based network should adapt existing protocols"? and "What do we gain in using specific protocols ?" This paper focuses on one of these features: the control of congestion. Despite the flow balance property of NDN that ensures that one request generates no more than one response on each link,

it does not guarantee the rate of Data and congestion may still occur.

Therefore the community have propose several solutions that fall into three types: end-to-end solutions that follow the footsteps of TCP, hop-by-hop ones that act as Internet load-balancer or queue manager (AQM), and some hybrid solutions that combine both approaches. In this paper, we propose to assess and to analyze one representative solution of each type: Interest Control Protocol (ICP) [2] for the end-to-end approach, Fast Pipeline Filling (FPF) [3] for the hop-by-hop approach and Practical Congestion cONtrol scheme (PCON) [4] for the hybrid approach.

We have implemented these protocols on the ndnSIM simulator [5]. We propose two topologies to compare their performances. All codes and environment parameters will be available¹ so new solutions might easily reproduce our results and compare them with their own.

The remaining sections of this paper are organized as follows. In Section II, we provide some background information and related work. Then, Section III presents the topologies and the different scenarios that we have used in this study. Section IV presents the evaluations of the three algorithms. In Section V, the paper drives a discussion on congestion control challenges and how application needs could be considered. Finally, Section VI concludes the paper and addresses ideas for future work. The contributions of this paper are summarized as follows:

- Implementations on the ndnSIM of congestion control protocols;
- Comparisons of the three main families of solutions;
- Guidelines on how to map congestion controls with applications needs.

II. BACKGROUND AND STATE OF THE ART

A. Forwarding Strategy Background

Multi-paths and multi-sources are native in NDN: nodes may have several choices to forward an Interest to Data producer. The algorithm that selects suitable paths is named *Forwarding Strategy* (FS). Many forwarding strategies have been defined and, in this study, we use and compare the Best Route strategy (BR) [6], Dynamic Request Forwarding strategy (DRF) [7], the Fast Pipeline Filling strategy (FPF)

¹<https://gitlab.tesa.prd.fr/athibaud/guidelines-codebase>

[3] and the Practical Congestion cONTrol strategy (PCON) [4]. FPF and PCON are particular forwarding strategies that propose to solve the congestion problem. These two strategies are thus described in Sec. II-B.

BR strategy is the standard forwarding strategy used in the NDN implementation [6]. It always selects the path with the smallest metric (traditionally, the hop count).

The purpose of the DRF strategy is to balance the use of the interfaces. Nodes monitor the number of Pending Interests (PI) on their interfaces in order to compute a weight. When a node wants to forward an Interest, it performs a weighted round robin to select the outgoing interface and then, updates its corresponding PI number and weight.

B. State of the Art in NDN Congestion Control

There are three families of methods to handle congestion issue in NDN.

The first type of solutions proposes an end-to-end approach. These solutions are similar to TCP. However, the congestion window is implemented at the consumer side, due to the receiver-driven property of NDN. Interest Control Protocol (ICP) is the more classical solution of this family. The consumer handles a congestion window for the content to be retrieved. The size of this window evolves using an Additive Increase Multiplicative Decrease mechanism (AIMD). When a Data packet arrives at the consumer, it increases the size of the congestion window additively. This additive increase corresponds to the congestion avoidance phase of classical TCP: the congestion window increases of one in a round-trip time (RTT). When congestion is detected by a timeout, the consumer decreases the window multiplicatively (divided by two here). The computation of the Retransmission Timeout (RTO) in ICP is the mean between the minimum and the maximum of the measured RTTs. Other notable solutions have to be mentioned: Multi-Path ICP (MPICP) [7] and Path-specified Transport Protocol (PTP) [8]. These two solutions follow the evolution of TCP to MPTCP [9] but for ICP. They both use a *route label* to identify the path used by the chunks and measure the RTT of each of the paths available. Furthermore, PTP uses the route label to force the forwarding of chunks on the known paths. Even if PTP changes the forwarding mechanism of NDN, it allows the use of TCP mechanisms like Fast Recovery. PTP uses a congestion window for each used paths and their increase are coupled by the Linked Increase algorithm to ensure a TCP-fairness. MPICP uses only one congestion window as in ICP. The difference is that the decrease of this window is probabilistic and depends on the measured RTTs of the current path. For this type of solutions, this paper only evaluates the popular ICP algorithm.

The second family of solutions is a hop-by-hop approach. As explained in the Sec. II-A, forwarding strategies can be used to handle the congestion problem. The Fast Pipeline Filling strategy is one of them. The purpose of this strategy is to use and fill all available paths as fast as possible. The

TABLE I: Summary of the described algorithms.

	Congestion Control		Forwarding Strategy
	End-to-End	Hop-by-Hop	
ICP	✓		
MPICP	✓		
PTP	✓		
FPF		✓	✓
PCON	✓	✓	✓
BR			✓
DRF			✓

node measures the mean RTT for each of its interfaces and computes a capacity defined as follow:

$$C = rtt * rate + queue_length \quad (1)$$

This capacity represents the number of data the node can handle from this interface. When an Interest needs to be forwarded, the node uses the interface with the smallest delay among the interfaces not completely filled.

Finally, the third kind of solutions is a hybrid approach that combine the previous solutions. In [4], PCON is designed: the end-to-end part is based on a congestion window at the consumer side whereas the hop-by-hop part is a forwarding strategy on each NDN nodes. We respectively name them PCON-CS and PCON-FS in this paper. PCON-FS uses the CoDel approach [10] to detect the congestion: nodes measure the sojourn time of each packet in their queue. If its mean during a given period is higher than a threshold, the interface is considered as congested. This interface marks the Data packets rather than dropping them in order to trigger an explicit adaptation from the upstream nodes. Other nodes receiving a marked Data shall not use this interface for this flow and prefers to use the other available interfaces instead. Initially, PCON-FS uses the path as defined in BR. With the marks it receives, the traffic is progressively load balanced on the other available paths. The PCON-CS consumer has the same behavior as ICP: a congestion avoidance phase and the RTO computation from RFC 6298 [11]. In addition, a marked Data is considered as a congestion notification and triggers a multiplicative decrease.

Every end-to-end algorithms are independent from the forwarding strategy. Thus in our evaluation, we will try different pairings as PCON-FS, BR or DRF.

III. SCENARIOS

In this section, we present the two topologies and scenarios that highlight the strengths and weaknesses of the studied solutions. For this purpose, the focus are on the following metrics: the application data rate of each consumer (the goodput), the fairness between the different users, the delay to retrieve each chunk and the sojourn time in the bottleneck queues. Our criteria to evaluate the different congestion algorithms are the goodput and the fairness between all the users. The delay and sojourn time metrics only help us to understand how those mechanisms work. For the fairness criterion, we do not use the popular Jain's fairness index since it focuses on giving

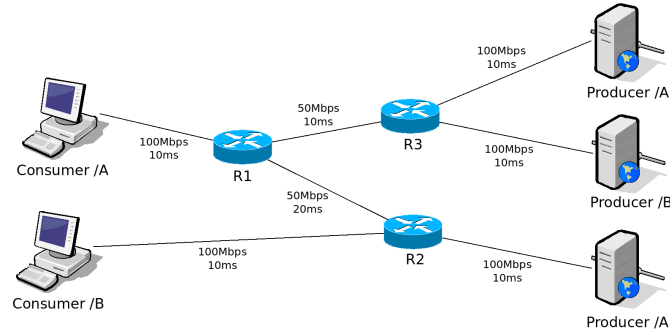


Fig. 1: Small scale topology

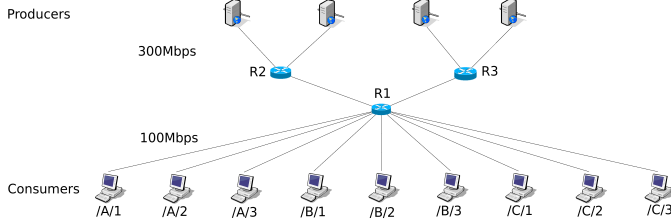


Fig. 2: Large scale topology

the same allocation to each user. We use the max-min fairness because we want that the users get a fair share of the available bandwidth without being limited by this fair share.

A. Topology configurations

The topology in Fig. 1 is a small-scale one with a few flows in order to understand clearly the behavior of the studied protocols in front of NDN features and opportunities. Indeed, a limited scale enables a more comprehensive study and if a solution does not take profits of a NDN feature or badly behave in this context, it will not be better in a larger topology. This topology offers many features. First, there are two paths to retrieve the data A. This allows us to evaluate these algorithms in an actual multipath scenario. Second, the path to upload the data B intersects with both paths to retrieve A. The first path of A and the path of B share the link between the node R1 and the node R3, and the Interests and Data are on the same direction. On the contrary, the second path of A and the path of B share the link between the node R1 and the node R2 and the Interests and Data are on different directions. This second property allows us to evaluate the impact of the different algorithms on the fairness of competitive flows. All the queues have a size of 120 layer-2 packets of 1500B. It represents 21 full data packets of 8400B. The size of A and B is 400MB and the producers use chunks of 8400B. Although this topology is simple, it allows us to highlight weaknesses of the studied solutions (cf IV). Furthermore, we conduct simulations on a topology where nine flows can compete at the same time (Fig. 2). We use those results to confirm our conclusions especially to see if good performances can support a larger use case.

B. Scenarios configurations

In our first scenario (Sce1), there is only the flow A. This scenario highlights the behavior of the algorithms when there is only one flow on the network and how multipath is handled.

TABLE II: Results Sce1.

	End-to-End Algorithm	Forwarding Strategy			
		BR	PCON-FS	DRF	FPF
Goodput (Mbps)	PCON-CS	40.2	52.9	64	40.2
	ICP	42	81.3	71.9	42.1
Delay	PCON-CS	64.3ms	69.4ms	74.6ms	64.3ms
	ICP	72.9ms	78ms	76.5ms	73.2ms
Sojourn time	PCON-CS	2.1ms	2ms	2.4ms	2.1ms
	ICP	10.3ms	6.5ms	3.9ms	10.2ms

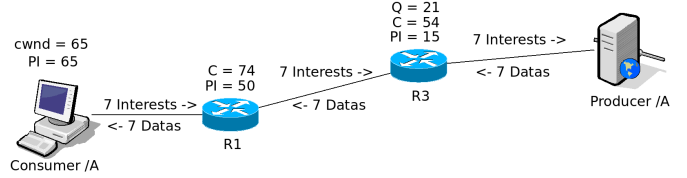


Fig. 3: FPF state before losses

Our second scenario (Sce2) shows how two flows (A and B) compete on the bottleneck link. The two flows compete on the R1-R3 link where they are in the same direction and on the R1-R2 link where they are in opposite direction. The flow B starts requesting data ten seconds after the flow A, so the first flow has already a stationary behavior. It allow us to evaluate conjointly the multipath efficiency and the fairness of the algorithms. Finally, our third scenario (Sce3) take place on the topology of Fig. 2. The nine consumers start at the same time, share the same producer three by three but are distinct flows. We distinguish two sub-scenario: a first (Sce3a) where the network has sufficiently capacity (links R1-R2 and R1-R3 have a capacity of 600Mbps) and a second (Sce3b) where the bottleneck happened between the nodes R1, R2 and R3 (links R1-R2 and R1-R3 have a capacity of 300Mbps).

IV. EVALUATIONS AND ANALYSIS

A. Results

The results of the Sce1 are summarized in table II. As the original RTO computation in ICP has showed really poor performance in our firsts simulations, we choose to use the classical computation, described in the RFC 6298.

Whatever the end-to-end congestion control used, ICP or PCON-CS, BR and FPF strategies propose similar performances. BR always chooses the same interface. FPF does the same but for a different reason: the computed capacity is not reached when a loss occurs. The consumer reduces its congestion window before R1 chooses to use the second path. The Fig. 3 shows the nodes used in this scenario, their PI number, their capacity, the current size of the bottleneck queue and the Interests and Data transiting on the links just before the first loss occurs. This state helps us to understand what goes wrong and why R1 has not yet chosen to use the second path. It is the state where Interests are paced at the Data rate of the bottleneck. This is why there are 7 Interests and 7 Data transiting on each links. The size of the congestion window of our consumer (ICP in this case) is 65. The PI number of R1 for the first path is 50 and of R3 is 15. The queue of the bottleneck (R3) is almost full and contains 21 Data packets. R1 computes a capacity of 74 Interests but the

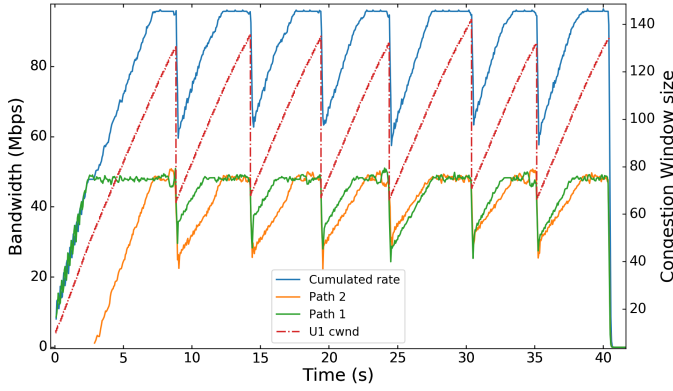


Fig. 4: Congestion window and rate for the consumer ICP

congestion happened when the PI number is 50. This is due to the queue length part in the formula 1. Without it, the capacity is approximately 50 and corresponds to the PI number. In this case, R1 would be able to use the second path and fully filling it too. Furthermore, the queue length has already (but indirectly) an impact on the capacity. Indeed, the increase of delay due to the queuing increases the capacity computed by R1. When Data arrives at R1 but the queue of R3 is still empty, the delay is smaller and the computed capacity is about 31 Interests. Only the bottleneck queue along the path should be considered to compute the capacity, and not queues of any other router. Otherwise, if the local queue is larger than the bottleneck one, the capacity will be oversized and will not really reflect the true capacity of the interface.

For these two forwarding strategies, consumer using ICP reaches a slightly better goodput than PCON-CS. Our results show that, with ICP, the queue of the bottleneck (R3) is slowly filling up while the congestion window increases, then drops packets. The consumer decreases the window when it detects the loss. In PCON-CS, R3 marks Data packets before the queue is completely full. No loss occurs but the consumer reduces its congestion window before the ICP consumer. PCON-CS is able to detect the congestion before a packet is lost. On average, the ICP consumer has a higher congestion window size than the PCON-CS consumer and thus a better goodput. The price to pay is losses and more variable delays. Indeed, the more the bottleneck queue is filled, the more the packets wait before been served. And when a loss occurs, the consumer needs to resend an Interest to get the Data (increasing the delay). The average end-to-end delay is about 72.9ms with the ICP consumer while it is only about 64.3ms with the PCON-CS consumer. The differences in the end-to-end delay are directly due to the sojourn time of the packets in the bottleneck queue. Indeed, a packet waits on average 2.1ms with the PCON-CS consumer while it waits 10.3ms with the ICP consumer.

PCON-FS shows some interesting results: Fig. 4 and 5 show the congestion window, the rate of the consumer and the rate on each path for the ICP and PCON-CS consumers respectively. The ICP consumer reaches the maximal rate of 100Mbps allowed by the topology and both queues of R2 and R3 are full when a loss is detected. PCON-FS slowly directs

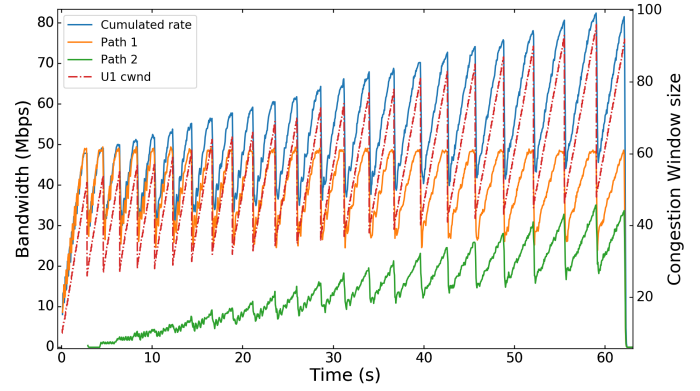


Fig. 5: Congestion window and rate for the consumer PCON-CS

Interest through the second path but only when the queue of R3 begins to be filled. Despite the losses and variable delay, this combination reaches an average rate of 81.3Mbps out of the 100Mbps available. Furthermore, no loss occurs until both paths are fully used. The marking mechanism of PCON-FS solution successfully avoids that a path is too much used until there is no other choice.

For the PCON-CS consumer, the second path is less used and never approaches the 50Mbps available. But, as for the BR strategy, the queues are never fully filled and the delay is not variable. The marks from R3 trigger the use of the second path by R1. But the swinging is too slow: at most 3% by mark. And marks have an effect on the consumer too. When it receives a mark, the consumer reduces its congestion window. The double effect of the marks induces a slow swinging on the second path and limits the goodput. The green curve on the Fig. 5 represents the bandwidth of the second path while the red one represents the congestion window. After each decrease of the congestion window, the rate of the second path increases. In comparison, on the Fig. 4, with the ICP consumer, the rate of the second path increases and reaches 50Mbps as soon as the first path is full. After each decrease, both paths are used the same.

The DRF strategy has different effect on the performances of the PCON-CS and ICP consumer. The ICP consumer has worse performances with DRF strategy than PCON-FS while it is the opposite for the PCON-CS consumer. As noticed earlier, the PCON-FS combined with the ICP consumer successfully uses both paths before a loss occurs. With the DRF strategy, both paths are used in an equivalent way from the beginning. However, due to the round robin, one of the path might be more used than the other and a loss can occur before both paths are fully used. It is still a great improvement in comparison of BR and FPF strategies that only use one path. For the PCON-CS consumer, the goodput is well improved with the DRF strategy in comparison of the PCON-FS. As for the ICP consumer, with the DRF strategy, R1 uses both paths at the beginning. While with PCON-FS, the second path is used but "crescendo".

The results of the Sce2 are summarized in tables III, IV and V. In this scenario, U2 begins 10 seconds after U1, U1 can

TABLE III: Results Sce2 for BR strategy.

			U1 only	U1 & U2	U2 only
ICP	U1	Path1	40.4Mbps	32.4Mbps	
		Path2	0.1Mbps	0.1Mbps	
		Total	40.5Mbps	32.5Mbps	
	U2	Total		12.1Mbps	41.3Mbps
PCON-CS	U1	Path1	37.8Mbps	25.9Mbps	
		Path2	Not used	Not used	
		Total	37.8Mbps	25.9Mbps	
	U2	Total		17.9Mbps	32Mbps

TABLE IV: Results Sce2 for PCON-FS.

			U1 only	U1 & U2	U2 only
ICP	U1	Path1	42.5Mbps	20.4Mbps	
		Path2	31.1Mbps ¹	43.4Mbps	
		Total	64.6Mbps	63.8Mbps	
	U2	Total		19.4Mbps	42.7Mbps
PCON-CS	U1	Path1	37.9Mbps	22.1Mbps	
		Path2	1.8Mbps ¹	18.4Mbps	
		Total	39Mbps	40.5Mbps	
	U2	Total		20.5Mbps	31.9Mbps

fetch the data from two sources and the first path has to share one link with the only path of U2 to his data source. Thus during the ten first seconds, there is only the U1 flow. Then, both flows are running. This phase is called the coexistence phase. Finally, U1 ends and only the U2 flow is running.

As in Sce1, the BR strategy use a single path. Using the ICP consumer, our user U1 has a mean rate of 32.4Mbps during the coexistence with the user U2. This last only has a mean rate of 12.1Mbps but is able to have a rate of 41.3Mbps when the user U1 is over. This unfairness comes from the burst losses of R3. Despite the fact that most of the dropped packets are from the U1 flow, only one window decrease is triggered on U1. Both U1 and U2 reduce their respective congestion window by two, independently of the number of losses detected. The congestion windows should finally converge to the same size, but this is only true if they increase at the same speed. However the end-to-end delay of the U1 flow is smaller than the U2 one and U1 increases his window faster than U2.

Using the PCON-CS consumer and during the coexistence, U1 has a mean rate of 25.9Mbps and U2 has a mean rate of 17.9Mbps. The marking mechanism on the node marks only one of the Data packet while the congestion is detected, using a marking timer to schedule the next mark. This Data packet is randomly picked when the congestion is detected. In this situation, only one of the stream is reduced. At the next congestion detection, the non reduced flow should be majority and thus would more likely be marked. This property promotes flow fairness.

The results for PCON-FS are very good. During the coexistence and with both consumers, the rate of the consumers are alike on the shared link. The difference between ICP and PCON-CS consumers (besides the losses and delay) is that the ICP consumer successfully uses the second path while the marks limit the use of the second path for the PCON-CS consumer. Fig. 6 shows how the coexistence is made on the shared path (green curve for U1 and red curve for U2) and

¹As seen in Fig. 6, this path is not used from t=0

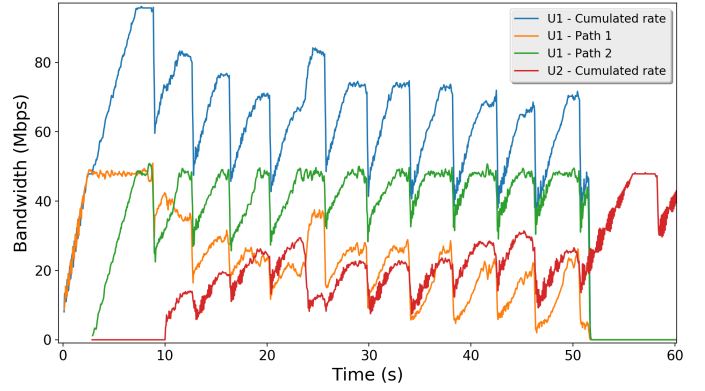


Fig. 6: Rates for the consumer ICP with PCON-FS - Sce2

TABLE V: Results Sce2 for DRF strategy.

			U1 only	U1 & U2	U2 only
ICP	U1	Path1	28Mbps	23.4Mbps	
		Path2	28.4Mbps	23.6Mbps	
		Total	56.4Mbps	47Mbps	
	U2	Total		17.4Mbps	42.3Mbps
PCON-CS	U1	Path1	25.2Mbps	22Mbps	
		Path2	25.4Mbps	22.3Mbps	
		Total	50.6Mbps	44.3Mbps	
	U2	Total		18.6Mbps	32.9Mbps

how the ICP consumer still uses the second path at it full capacity (orange curve).

For the DRF strategy, both consumers act the same. Due to the balance use of the available path, both paths are used at the bottleneck rate. Despite the fairness of this combination, the second and non congested path is not fully used.

Table VI represents the results of scenarios Sce3a and b. For the unconstrained case, each flow can have 100Mbps. This scenario shows how the end-to-end algorithm behave when there is no competition on the links. As seen earlier, PCON-CS has lower rates than ICP. This is due to the early detection of the congestion (via the marks) and allow to avoid losses on the congested links. For the constrained case, the nine flows have to share 600Mbps. Since the competition between the flows happened on both the paths of the topology, DRF does not show the inefficiency seen previously. However, the combination PCON-CS and PCON-FS show a great unfairness for the three flows that look for the Data A. Since this Data is available on both the path, the PCON-FS slowly load balance the traffic of these flows. But the others flows occupy both the paths to and the A flows get lowers rates.

B. Synthesis

As noticed in our evaluations, ICP consumer detects the congestion with timeouts and thus is based on losses. The PCON-CS consumer is based on congestion marks added by

TABLE VI: Mean rates (Mbps) on larger topology

End-to-End Algorithm	Forwarding Strategy	Unconstrained case (Sce3a)	Constrained case (Sce3b)
ICP	PCON-FS	88.383 ±1.13573	65.331 ±6.83087
	DRF	88.965 ±1.1493	66.248 ±8.0037
PCON-CS	PCON-FS	76.659 ±0.66441	59.151 ±19.54676
	DRF	76.76 ±0.55762	61.622 ±2.55499

the nodes that have too long waiting service for the packets. With the ICP consumer, the queue of bottlenecks is gradually filled, end-to-end delays increase and finally losses occur. With the PCON-CS consumer, bottleneck nodes mark the Data before the queues are completely full, the end-to-end delays is stable and losses are prevented. In terms of goodput, the consumer ICP shows the best performances. Indeed, as it waits for a loss to occur before decreasing his congestion window, the window mean size is higher than with the PCON-CS consumer.

For the hop-by-hop algorithm, FPF strategy shows some design issues with the computation of the capacity of the interface. The considered queue should be the bottleneck one, not the local one since it is this queue that will be filled up eventually. Furthermore, as the results showed us, the length of the bottleneck queue is already taken into account with the increase of the delay induced by the queuing delay. Indeed, the more the queue is full, the more packets wait before been served and the higher the computed capacity is. The PCON-FS strategy shows really good results when combined with the ICP consumer. The nodes are able to efficiently use the congestion marks to lead traffic on both paths. After an initialization phase, both paths are used in an equivalent way. But with the PCON-CS consumer (the original combination), the congestion marks have an effect on the forwarding strategy and on the consumer. This double effect leads to a poor utilization of the second path and a smallest overall goodput.

V. CHALLENGES

NDN is a data-oriented network layer. One obvious application for this new paradigm is a file downloading application. NDN must have good performances for this data-oriented applications. From a user point of view, only the goodput is important to optimize for this application. In terms of goodput, the best combination is to use ICP as the end-to-end congestion control algorithm and PCON-FS as the forwarding strategy on each node. Furthermore, our results show good fairness when several flows are competing on a same link.

If the targeted application is sensible on losses or variable delay, using ICP as the end-to-end congestion control might not be the optimal solution. Indeed, ICP detects the congestion when a loss occurs. It induces the filling of the bottleneck queue and the increase of the end-to-end delay. An approach similar to the PCON algorithm could be a solution. PCON is based on the CoDel AQM principle. The congestion is detected directly by the bottleneck queue and before it is completely full. Most of the losses are prevented and the delay stays stable at least on each path. Indeed, the multipath data retrieval induced by NDN naturally makes the overall end-to-end delay variable. A critical delay-sensitive application might need to use a new congestion control mechanism that ensures the use of only one path. With existing solutions, the best combination is to use PCON-CS as the end-to-end algorithm and DRF as the forwarding strategy on each node. This combination shows a good fairness property too but all the sub-paths from a DRF node will have the same rate (the bottleneck one), even if more

bandwidth is still available. This is due to the inherent balance used of the paths of the DRF strategy.

The cooperation between an end-to-end mechanism and a forwarding strategy is a good idea to solve the congestion issue, as in ECN related works in TCP/IP networks. Even if the PCON proposition as a whole does not present goods results, it is a lead that needs further investigation. Its main weakness is that the congestion marks trigger adaptation on the consumer and on the nodes on the path. One of these adaptations could solve the congestion problem but the other might only reduce the overall rate. A finer-grained communication between the nodes and the consumer needs to be designed to do the rate adaptation at the right location. A possible solution would be to let the node solve the congestion problem by using other non congested paths. If all paths are congested, the node notifies the upstream node but this notification is not automatically forwarded to the consumer. The congestion is reevaluated on each node until it is solved.

Finally, the fairness of these algorithms is very important. Without any counter-argument, all the users should have the same share of the bandwidth. As multipath communications are enabled in NDN, a user might benefit of an increased bandwidth due to the multiple locations of its targeted data while an other user might only have one path available. It is the case in our Sce2 and we consider the fairness on the competing link only. However, we could have chosen the competing node instead: a fair share would have be to use one link for the first user, and the other for the second user alone. In this case, they would get approximately the same overall bandwidth. The extreme case is to consider the whole network but this induces many problems such as the limited knowledge of the nodes, or the overhead to communicate all information to perform the fairness. Nevertheless, the link sharing seems to be a good criteria to evaluate the fairness of the algorithms.

VI. CONCLUSIONS

In this paper, we study and compare several approaches of congestion management in NDN. As expected, no single solution fits all cases and applications. The congestion detection method could induce losses and high variation in the end-to-end delay. Thus, the selected solution should depend on the targeted goal. Our simulations, conducted on small and large topologies, show that some trends can be raised. By providing some guidelines, we hope to help the NDN community in search of the best congestion control algorithm. Our conclusions are based on reproducible simulations so the community can compare their results with our own. Our analysis has shown that receiver-driven communication and hop-by-hop forwarding mechanisms need to cooperate in order to efficiently resolve the congestion issue and optimize the goodput of the final user. In our future work, we plan to design a new solution where each node will try to solve the congestion problem when detected (locally or by notification) and notify the upstream node if it cannot. We expect it will successfully use the multiple paths available while avoiding the congestion and staying fair between users.

REFERENCES

- [1] L. Zhan, A. Afanasyev, J. Burke, V. Jacobson, kc claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang. Named data networking. *SIGCOMM Comput. Commun. Rev.*, 44(3):66–73, July 2014.
- [2] G. Carofiglio, M. Gallo, and L. Muscariello. Icp: Design and evaluation of an interest control protocol for content-centric networking. In *2012 Proceedings IEEE INFOCOM Workshops*, pages 304–309, March 2012.
- [3] A. Detti, C. Pisa, and N. Blefari Melazzi. Modeling multipath forwarding strategies in information centric networks. In *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 324–329, April 2015.
- [4] K. Schneider, C. Yi, B. Zhang, and L. Zhang. A practical congestion control scheme for named data networking. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking, ACM-ICN '16*, pages 21–30, New York, NY, USA, 2016. ACM.
- [5] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang. ndnSIM 2: An updated NDN simulator for NS-3. Technical Report NDN-0028, Revision 2, NDN, November 2016.
- [6] A. Afanasyev, J. Shi, B. Zhang, L. Zhang, I. Moiseenko, Y. Yu, W. Shang, Y. Li, S. Mastorakis, Y. Huang, J. P. Abraham, E. Newberry, S. DiBenedetto, C. Fan, C. Papadopoulos, D. Pesavento, G. Grassi, G. Pau, H. Zhang, T. Song, H. Yuan, H. B. Abraham, P. Crowley, S. O. Amin, V. Lehman, M. Chowdhury, and L. Wang. Nfd developer’s guide. Technical Report NDN-0021, Revision 7, NDN, October 2016.
- [7] G. Carofiglio, M. Gallo, L. Muscariello, M. Papalini, and S. Wang. Optimal multipath congestion control and request forwarding in information-centric networks. In *2013 21st IEEE International Conference on Network Protocols (ICNP)*, pages 1–10, Oct 2013.
- [8] Yuhang Ye, Brian Lee, Ronan Flynn, Niall Murray, and Yuansong Qiao. PTP: path-specified transport protocol for concurrent multipath transmission in named data networks. *CoRR*, abs/1802.02828, 2018.
- [9] Alan Ford, Costin Raiciu, Mark J. Handley, and Olivier Bonaventure. TCP Extensions for Multipath Operation with Multiple Addresses. RFC 6824, January 2013.
- [10] K. Nichols, V. Jacobson, A. McGregor, and J. Iyengar. Controlled Delay Active Queue Management. RFC 8289, January 2018.
- [11] M. Sargent, J. Chu, Dr. V. Paxson, and M. Allman. Computing TCP’s Retransmission Timer. RFC 6298, June 2011.