# Reactivity Enhancement of Cooperative Congestion Control for Satellite Networks

Adrien Thibaud\*, Julien Fasson\*, Fabrice Arnal<sup>‡</sup>, Renaud Sallantin<sup>‡</sup>, Emmanuel Dubois<sup>§</sup> and Emmanuel Chaput\*

\*TéSA Laboratory, firstname.name@tesa.prd.fr

\*ENSEEIHT/IRIT/University of Toulouse, firstname.name@enseeiht.fr

<sup>‡</sup>Thales Alenia Space, firstname.name@thalesaleniaspace.com

<sup>§</sup>Centre National d'Études Spatiales, emmanuel.dubois@cnes.fr

Abstract—The new paradigm of Information Centric Network (ICN) proposes a shift from the host-centric model to a contentcentric model. This approach, especially well suited to the current Internet's usage, is promising for Satellite Networks. In particular, Named Data Networking (NDN) architecture seems to be a great candidate: it gathers the benefits of Content Delivery Networks (CDN), Peer-to-Peer networks (P2P) and HTTP in the network layer. In this study, we propose to compare the performances of TCP-like congestion control algorithms and our new Cooperative Congestion Control (CCC) approach. CCC is a pace-based multipath and multi-flow aware congestion control. We evaluate those algorithms with simulations on a topology where we place the satellite link on different positions. We show that CCC outperforms window-based algorithms but has still some drawbacks. We thus proposed an enhancement of CCC that corrects the flaws by increasing its reactivity. Simulations results show that the performances on terrestrial scenarios are also enhanced.

Index Terms—ICN, NDN, QoE, Max-min fairness, Multipath flows, Congestion Control, Satellite Networks

#### I. INTRODUCTION

Information Centric Network (ICN), the new paradigm of the network layer, proposes to shift from host-centric communications to communications where the user retrieves the requested content independently of its location in the network. Indeed, the current Internet's usage has evolved a lot since its creation. Today, it is more focused on how fast a content is retrieved than where it comes from. Geostationary satellite networks do not escape this trend and face a specific challenge: the long delay induced by the satellite's high altitude degrades significantly the Quality of Experience (QoE) of the users, especially for interactive applications which require multiple exchanges of packets. Conversely, for content retrieving based applications, only the completion time matters for the users' QoE and the satellite delay is only added once for a small percentage of the overall completion time. This additional delay has then a lesser impact on the OoE.

Content Delivery Networks (CDN), which are a first flavor of ICN, have shown promising results on satellite networks [1]. Named Data Networking [2] (NDN) is the most popular ICN architecture and could offer an opportunity to increase even more the QoE of satellite users than the CDN solution. Indeed, NDN gathers the benefits of CDN but also of Peer-to-Peer networks (P2P) and HTTP. As in CDN, its nodes are able to store the content and to reserve it when requested. NDN is a receiver-driven architecture: the communication is initiated by the user, the Consumer in NDN terminology, when it sends Interest packets. Then, servers or Producers in NDN send Data packets which follow the reverse path of the Interests. The Interests have a similar purpose of the GET messages of HTTP but at a network level. As in P2P, multi-producers and multi-paths are used by the network to lead requests and responses between consumers and producers. The multiproducers property is emphasized by the caching capability of the nodes. Indeed, this ability is accentuated by the intrinsic security of the Data packets. It does not depend on the end points and any node is able to check the authenticity of a Data packet. It can then easily reuse stored Data packets for an other consumer request. For the same reason, nodes are able to multicast Data packets when multiple consumers request them at the same time.

The next step is then to study in more depth the performances of NDN on satellite topologies. The satellite link possesses some particular properties: due to its high altitude, the satellite has a long propagation delay and a very large coverage (about a third of the globe for geostationary satellites). In classical TCP/IP networks, the important bandwidth delay product jeopardizes the performance of the windowbased congestion control algorithms. This is why we choose to focus on the congestion control in Named Data Satellite Networks. In this paper, we compare classical NDN windowbased congestion control mechanisms' performances with our solution Cooperative Congestion Control [3] (CCC), all being initially designed for terrestrial networks. After showing the superiority of CCC, we propose an enhancement of its reactivity when the balance of flows changes. Furthermore, NDN provides the opportunity to redefine completely the way to solve problems and we show here that these approaches can have better results than former ones.

The remaining sections of this paper are organized as follows. In Section II, we survey the related work. Then, in Section III, we discuss about the different approaches that could tackle our challenge. In Section IV, we present the principles of CCC, our previous solution for terrestrial networks. Section V presents the evaluations conducted on satellite topologies. In Section VI, we propose an enhancement of CCC and the new simulations demonstrating its benefits. Lastly, the paper is concluded in Section VII.

#### **II. RELATED WORKS**

Only a few studies have been made on ICN implementation over satellite networks. Among these, there is the ESA ARTES study ' $\varphi$ SAT'. For example, [4] and [5] focus on the PURSUIT architecture [6] on hybrid satellite and terrestrial networks. The authors study the performance of ICN for specific scenarios and applications, such as broadcast IPTV or machine-tomachine smart transport. In [7], the authors study the general features of ICN architecture and how they interact with a satellite link. In particular, they explain that the hop-by-hop approach for the congestion problem should: (a) help with the heterogeneity of physical layers across the end-to-end paths, (b) grab the opportunity to define an unique solution working on all type of network and removing the need of proxies (such as Performance-Enhancing Proxies) and (c) exploit the multipath and multi-source in order to have an efficient use of the satellite resources. In [8], the authors compare the multimedia content distribution over satellite networks with IP and NDN stack. They claim that NDN provides a better throughput and end-to-end delay.

In term of congestion control in NDN, to the best of our knowledge, there are only a few propositions for satellite networks. This is why we focus on classical congestion control algorithms. In a previous study [9], we have highlighted two combinations of end-to-end and hop-by-hop solutions that are the most efficient to solve the congestion issue while still using suitably the network resources. Those two combinations are "ICP + PCON-FS" and "PCON-CS + DRF" which we will compare to our proposition CCC in the remainder of this paper. ICP stands for Interest Control Protocol [10]. It is a receiverdriven adaptation of TCP for NDN. The consumer manages a congestion window of Interests that uses an Additive Increase Multiple Decrease algorithm (AIMD). We couple it with the forwarding strategy of PCON [11]. PCON is an hybrid solution with an end-to-end and a hop-by-hop part. We call the hopby-hop part PCON-FS for forwarding strategy and the endto-end part PCON-CS for consumer side. The PCON-FS use the principle of the Active Queue Management (AQM) CoDel [12] in order to estimate if a queue is congested or not. If it is, the node marks the Data packet with a congestion tag. When the forwarding strategy receives marked Data, it slowly loadbalances the traffic away of this path. The PCON-CS is an adaptation of TCP too. The most salient difference with ICP is that it will also trigger a multiplicative decrease when it receives a marked Data. We have shown in [9] that it is better to use those two parts (Forwarding Strategy and Consumer Side) separately. Indeed, they both take action on a congestion notification and thus globally overreact to this event. For the second combination, we use the Dynamic Request Forwarding strategy [13]. The purpose of DRF is not congestion control but load balancing at local node level. The node computes a weight for each of their interfaces by counting the pending interests. To forward an Interest, the node performs a weighted round robin to choose the outgoing interface. The final goal is to use the available paths at the same proportions.

## III. DISCUSSIONS ON NDN CONGESTION APPROACHES

In this section, we discuss about the different approaches in order to control the congestion in a satellite network, using the NDN stack. As we have seen in the previous section, a lot of solutions used in terrestrial networks are TCP adaptations. However, window-based approaches struggle to use efficiently geostationary satellite links. Indeed, a long delay implies a slow increase of the congestion window. Furthermore, the high bandwidth of the satellite link coupled with the multiple decrease of the window induces an important decrease of the user goodput. Eventually, those two factors combined lead to long and inefficient fluctuations of the congestion window. To mitigate that, satellite operators usually use Performance-Enhancing Proxies [14] (PEPs). The principle is to split the TCP communication into multiple connections. The satellite part can then use specifics parameters set by the operator. The good point is that this is transparent from the end users point of view: no need to use a specific version of TCP. An adaptation of this method has to be considered, since window-based mechanisms are also used in NDN. Due to NDN's receiver-driven style, this approach seems difficult to adjust. Indeed, in TCP's PEPs, proxies spoof the connection by sending acknowledgment in advance. The purpose is to fake the RTT computed by the sender and then to increase faster the congestion window. In NDN, there is no acknowledgment. Instead, in window-based mechanisms, the Data packets take their roles. Thus, a PEP in NDN will not be able to send the Data packets before they receive them. One solution could be to send fake Data packets in order to trigger a faster increase of the congestion window and when the real Data packet arrives, send it to the consumer. But it introduces too many conflicts with the design of NDN. NDN defines the flow balance as the fact that one Interest packet induces at most one Data packet. Furthermore, nodes on the path are supposed to erase the reverse path information of an Interest as soon as the corresponding Data packet has taken this reverse path. Our fake and real Data packets break those two NDN properties. This is the first reason why we decide to have a different approach. The other reason is that NDN redefines completely the network stack and then allows everyone to propose new types of solution. So, we take the chance to design our own solution, using the main principles of NDN as flow balance, hop-by-hop capacity and bringing up a cooperation between all the nodes of the network. This proposition is presented in the next section.

#### IV. COOPERATIVE CONGESTION CONTROL

In a previous work [3], we introduced Cooperative Congestion Control (CCC), which is initially designed for terrestrial networks. Here, we present the CCC architecture and main objectives concisely but completely, in order to understand the results and the enhancements we have done for satellite networks. It is based on a per-node approach, works with the pace of the Interests and has three main principles: (i) a cooperation between all the nodes of the network, (ii) a



Fig. 1: Node context

local supervision of the transmitting queues and (iii) a smart allocation of the bandwidth for each active flows.

The cooperation part of our algorithm consists in the exchange of control data (in-band signaling) between the nodes. These exchanges aim to provide the throughput objectives and constraints of each flow to the nodes. They are represented on the Fig. 1. In this figure, we focus on one flow and one node. The requests arrive from the interface 0 and are forwarded to interfaces 1 to n (Upstream). The responses follow the reverse direction (Downstream). From Upstream, the node receives the pace objective of each flow  $(p_{i\_}obj)$ . It is implemented as a local objective pace for the flow as it could have been divided earlier in other paths. The node has then to divide this objective between its own n available paths  $(p_{o,k\_}obj, \forall k)$ , such as:

$$p_{i\_}obj = \sum_{k=1}^{n} p_{o,k\_}obj \tag{1}$$

Eq. (1) represents the distribution and conservation of the pace objective in the considered node. No objective is created by the nodes and, on the global scale, the objective sent by the consumer is equal to the sum of all the objective received by the producers.  $p_{o,k}$ \_obj represents the outgoing objective pace for the interface k. From Downstream, the node receives the bandwidth constraint for each flow  $(p_{i,k}\_max, \forall k)$ . It represents the aggregated constraint pace for the flow and all the paths from the interface k. The node has then to aggregate these n constraints for the requesting interface  $(p_o\_max)$ , such as:

$$p_{o}max \le \sum_{k=1}^{n} p_{i,k}max \tag{2}$$

Eq. (2) represents the aggregation of the constraints and is not an equality because of the potential local congestion issue.  $p_o_max$  represents the local and outgoing constraint pace of the node.

The local supervision part of our algorithm consists, as it name suggests, in monitoring of transmission queues. Periodically, nodes check the status of these queues by measuring the number of dropped packets during the period. Queues are considered congested when, at least one packet has been dropped. Furthermore, nodes estimate the load of the link



Fig. 2: Testing topology

using the constrained pace and the size of the data packets of each active flows:

$$bandwidth\_estimation = \sum_{flowF} p_o^F\_max * data\_size^F \quad (3)$$

It can then deduce the spare rate for the interface, supposing that it knows the capacity of the link.

With all this information, CCC is able to use efficiently the multipath capacities of NDN while avoiding congestion. First, when a congestion is detected, the node reduces the outgoing constraint pace of each active flow by ten percent (implementation choice). Then, when a spare rate is available, the node shares it with all the active flows by increasing equally their outgoing constraint pace (Eq. (2) is still respected). Finally, the node uses the incoming constraint pace  $p_{i,k}$  max to decide how much each path is really used by a flow. Indeed, it sets the outgoing pace of the Interests  $p_{o,k}$  such as:

$$p_{o,k} \le p_{i,k} max, \forall k \tag{4}$$

A weighted round robin (implementation choice) is then used to select which interface transmits the current incoming Interest and a pacing is done on each output interface with the corresponding output pace observing by Eq. (4).

CCC is proposed as a framework and the three principles can be implemented in multiples ways. The principles are divided in six different algorithms and a first implementation of them is available in [3]. They can be freely modified in order to enhance the performance of the solution. Our purpose in this paper is to compare the performances of CCC with windowbased algorithms on satellite networks and propose a potential improvement on the implementation of CCC.

#### V. EVALUATIONS ON SATELLITE NETWORKS

In this section, we propose to compare the performances of CCC and classical solutions on a NDN topology including a satellite link. Our evaluations are done on the ndnSIM simulator [15] and we have simulated the satellite link with a link with a one-way delay of 300 ms. All the codes and environment parameters are available<sup>1</sup> in order to be able to reproduce our results. We choose to study the small scale

<sup>&</sup>lt;sup>1</sup>https://gitlab.tesa.prd.fr/athibaud/ccc-codebase/tree/reactivity\_enhancement





topology presented in Fig. 2. This approach allows to highlight the properties we want to study while keeping the necessary simplicity to avoid any possible wrong interpretations. The Consumer A has two paths to retrieve its content (multipath retrieving). Furthermore, Consumer B shares two links with the flow of Consumer A (concurrent retrieving): one where both the flows are on the same direction, between R1 and R3, and an other where the flows are on opposite direction, between R1 and R2. Consumer B starts requesting content ten seconds after Consumer A. It allows us to evaluate the fairness of the bandwidth share between the two flows. We conduct multiple tests depending on the satellite link usage:

- a reference case, without any satellite link;
- a consumer access case, where the satellite link is used as the access network of the Consumer A (between Consumer A and R1);
- a backbone case, where the satellite link is in the backbone network (between R1 and R3);
- and a producer access case, where the satellite link is the access network of the Producer A1 (between R3 and Producer A1).

# A. Reference scenario

Fig. 3 shows the rate of flows A and B when there is no satellite link in the topology. It is a reference where we can see the periodic oscillations of the congestion window-based solutions. Our solution has a stable rate which increases or decreases only when a flow begins or ends. All the solutions share the bandwidth of the congested link (R1-R3) equally between the two flows when they are both active.

#### B. Consumer Access Case

Fig. 4 exhibits the rate of flows A and B when the satellite link is in the access network of the Consumer A. This means

Fig. 4: Satellite link on Consumer A access network (Consumer A - R1)

that the Consumer B does not use the satellite link to retrieve its content. That is why the rate of the flow B is not impacted by the addition of a satellite link in the topology. Aside from CCC, the consumer B retrieves its content even faster because the flow A is not able to obtain a good rate. The combinations "ICP + PCON-FS" and "PCON-CS + DRF" suffer from the long delay induce by the satellite in the access network. It impacts the induced Round-Trip Time (RTT) of both paths and the window-based algorithms of ICP and PCON are never able to reach a good throughput. Our solution, CCC, is not affected by the satellite link delay and shows equivalent performances than in previous case. The consumer A is able to get all the throughput available when it is alone and shares the congested link equally when the flow B is also active.

## C. Backbone Case

Fig. 5 presents the results of the scenario where the satellite is on the backbone link. In this case, only one of the paths of the Consumer A is concerned and the second path has an usual RTT of a terrestrial network. The Consumer B also undergoes the satellite delay. As a result, the rates of both flows never take off with the window-based solutions. An interesting result is that the Consumer A with "PCON-CS + DRF" is able to use the non-satellite path to increase its congestion window and then achieves a better, but still low, throughput than "ICP + PCON-FS" solution. This is due to the load-balancing property of DRF that tends to use the two paths available equally while PCON-FS only balances the load on the second path when a congestion occurs. CCC is again able to fully use the bandwidth but when the Consumer B begins, both the flows take time to stabilize their rate. This is because the congested link is here the satellite link. In CCC, nodes monitor the congestion, take actions and notify their



Fig. 5: Satellite link on Backbone (R1 - R3)

neighbors. In this case, the notifications and the reactions are slower. Thus, when a new flow starts, the convergence time is longer and significant. However, the overall rates of CCC are still far better than the window-based solutions.

#### D. Producer Access Case

Fig. 6 shows the rates of flow A and B when the satellite link is on the access network of Producer A1. In this configuration, the flow A suffers from the satellite delay on only one of its paths and the flow B does not suffer. Therefore, the flow B is not impacted by the satellite and has even slightly better throughput because of the low performances of the flow A. CCC is not impacted either by the satellite delay, even for the flow A. It shows again the best throughput and is able to use the network at its full capacity. For the window-based solutions, the flow A shows poor performances too. But in this case, "ICP + PCON-FS" solution is helped by the flow B. Indeed, flow B fills the shared link and triggers the load balancing of PCON for the flow A. Even if the load-balancing proposed by PCON is slow, it eventually enables to fully use the non-satellite path. When this path is fully used (around 60s), we can see the slope of the rate decreasing (but still positive). Indeed, using a window-based mechanism, this slope depends on the delay of the path used. For the first ten seconds, only the satellite path was used. The speed of the increase is then long and the increase itself small. When the second flow begins, it congests this path and triggers a load balancing for the first flow on its non-satellite path. Both paths are now used and the increase of the window and the load-balancing of PCON finally achieve the filling of the non-satellite path around 60s. The increase of the window now only affects the satellite path (with a longer delay) and this is why the slope decreases. The overall rate is still low but much lower than the two other cases. For the "PCON-CS + DRF" solution, the



Fig. 6: Satellite link on Producer A1 access network (R3 - Producer A1)

flow A has the same behavior as with the satellite link on the backbone network since both these scenarios offer a short and a long delay paths. The difference here is that the flow B performs a good rate and triggers decreases on the flow A.

#### E. Conclusions

We conduct several scenarios depending on the role of the satellite in the network. The window-based solutions are never able to perform well on a satellite path. Even when a condition is beneficial, when a second non satellite path is available or when an other flow triggers a favorable load-balancing, it still achieves a lower throughput than the available one. Between the two window-based combinations we have studied, none is globally better than the other since the issue comes from the same base: the TCP-like congestion window. Indeed, such algorithms are designed for terrestrial networks and rely on a short RTT to perform well. Otherwise, CCC clearly outperforms window based solutions in every situation. However, one limit appears when the congested link is the satellite link. It leads to a longer convergence time when a new flow begins but CCC is still able to outperform window-based solutions.

### VI. REACTIVITY ENHANCEMENT OF CCC

As we have seen in the previous part, CCC has some issues in the case where the congested link is the satellite (Fig. 5). When the flow A is alone, it fully uses both paths, including the satellite path, with no issue. But when the flow B begins, the satellite link has to be shared between the two flows. On the others cases (non-satellite case included), there is a short convergence period and then the congested link is fully used and shared equally between the two flows. Here, there is a significant period where both flows try to retrieve at full rate and most of the Data are lost on R3. The two flows achieve lower throughput than the fair share on this link until the problem is eventually solved. Our purpose is to understand the root cause and to propose an enhancement of CCC in order to address it. Indeed, the design of CCC is naturally modular and is divided in six main algorithms [3]. This provides an easy way to enhance even more its performances, which is desirable in some particular cases.

When the node R3 detects the congestion, it decreases the outgoing constraint pace  $(p_o_max)$  of flows A and B. The Algo. 5 of CCC specifies to decrease those paces by ten percent, which was set as arbitrary value. In this case, this global decrease is not enough to solve the congestion. Multiple supervision periods and successive decreases are necessary to fix the congestion issue. This is the first part of our problem. But it also happens in the other scenarios, when the satellite link is placed elsewhere or is absent, and the problem does not occur. This is due to the second part of our problem. In the case the satellite link is the congested link, the notifications of R3 (in the form of the reduced outgoing constraint paces) take a longer time to arrive to R1. They are then applied by R1 but the effect on the incoming pace in R3 happens again around 300ms later. The reaction time of one decrease is then of 600ms. Since there is the need of multiple and successive decreases to resolve the congestion issue, the convergence time is significant.

The propagation time of the notification is independent of CCC and cannot be reduced. We must then tackle the first part of the problem. Instead of blindly decrease the constraint of the active flows, our proposition is to reallocate completely the bandwidth between them when a congestion occurs. To this end, we have redefined the queue supervision and pace reduction algorithms (Algo. 3 and 5 of CCC [3]). Now, when a congestion is detected, the node resets all the constraints of the interface, through the new pace reduction algorithm (Algo. 5). Then, they evaluate the available bandwidth to ninety percent of the link capacity, through the new queue supervision algorithm (Algo. 3). Finally, they distribute it between all the active flows, through the unchanged pace augmentation algorithm (Algo. 6). We decide to only

Modified Algorithm 3 Queue supervision	
Get queue status	
if congestion then	
Reduce the pace of each active flow (Algo. 5)	
Consider only 90% of the link capacity	
end if	
Evaluate available bandwidth	
Distribute it equally between active flow (Algo. 6)	
	_

	-
for	each active flow do
R	eset outgoing constraint
end	for

Modified Algorithm 5 Pace reduction

distribute ninety percent of the link capacity in order to help



Fig. 7: Comparison between CCC and Enhanced CCC

clear the transmission queue. When the congestion is over, the remaining ten percent are then distributed to the active flows and all the bandwidth is used. This new reaction to the congestion issue is done in only one supervision period and should increase the reactivity of CCC.

Fig. 7 shows the results of CCC with and without this enhancement. Sub-Fib. 7a and 7c represent the rate of flow A and B with CCC, respectfully and Sub-Fig. 7b and 7d represent the rate of flow A and B with our enhancement. The figures only show the cohabitation phase of the two flows (from 10s to around 24s). As we highlighted earlier, former implementation of CCC takes a long time to react when the congested link is the satellite link (backbone case). With our reactivity enhancement, the solution is able to reallocate the bandwidth as soon as the congestion is detected. Then, it only takes the propagation time to the next node to be active. On Sub-Fig. 7b and 7d, when the flow B begins (at 10s), there is a short phase of convergence for all the scenarios, which can last for 6s in the worst case. Our proposition enhances the results: the worst case is up to a convergence phase of less than 2s. Even when no satellite link is present, this phase is shortened by this latest proposition.

### VII. CONCLUSIONS

In this paper, we study the behavior of window-based congestion algorithms and our solution, Cooperative Congestion Control, on Named Data Satellite Networks. The windowbased solutions are never able to increase the size of their window enough to reach the real available throughput. Besides that, CCC outperforms the other solutions regardless of satellite position in the topology. Its pace-based approach shows few defects and allows the users to reach the maximum throughput. However, the hop-by-hop notifications of CCC come with the cost of a delay in the congestion resolution when the bottleneck link is the satellite link. Coupled with the current way CCC reacts to a congestion event (decrease by ten percent all active flows), we notice a long convergence phase when a new flow begins. We proposed an enhancement that allows CCC to increase its reactivity in such cases and offers a resolution of the congestion in only one supervision period. Simulation results show that the performances are better in all the cases, even on terrestrial scenarios. We also highlight in this paper that NDN provides the possibility to propose new and original approaches. Former solutions may be adapted but we have here an opportunity to rethink the problems and the solutions. This is the reason why we propose CCC as a modular framework, through its numerous subalgorithms. In future works, we plan to take advantage of this modularity to tackle the fairness problem. Indeed, CCC provides a fairness on the shared link: each flow gets a fair share of the bandwidth of the link. But some of these flows may have other possibilities to reach producers and then get an overall better throughput. We think that a node fairness can be easily reachable since it has a bigger point of view on the rate of the flows. Ultimately, a network fairness is considered but is a more challenging problem.

#### References

- Adrien Thibaud, Julien Fasson, Fabrice Arnal, David Pradas, Emmanuel Dubois, and Emmanuel Chaput. QoE enhancements on satellite networks through the use of caches. *International Journal of Satellite Communications and Networking*, 36(6):553–565, 2018.
- [2] L. Zhan and al. Named data networking. SIGCOMM Comput. Commun. Rev., 44(3):66–73, July 2014.
- [3] Adrien Thibaud, Julien Fasson, Fabrice Arnal, Renaud Sallantin, Emmanuel Dubois, and Emmanuel Chaput. Cooperative Congestion Control in NDN. In 2020 IEEE International Conference on Communications (ICC): Next-Generation Networking and Internet Symposium (IEEE ICC'20 - NGNI Symposium), Dublin, Ireland, June 2020.
- [4] Christopher N. Ververidis, Pantelis A. Frangoudis, Yannis Thomas, Vasilios A. Siris, George C. Polyzos, Ilias Andrikopoulos, Fabrice Arnal,
- [12] K. Nichols, V. Jacobson, A. McGregor, and J. Iyengar. Controlled Delay Active Queue Management. RFC 8289, January 2018.

Cédric Baudoin, and Maria Guta. Demonstrating information centric networking over integrated satellite / terrestrial networks. 2012.

- [5] Maria Guta, C. Ververidis, A. Drougas, I. Andrikopoulos, V. Siris, G. Polyzos, F. Arnal, and C. Baudoin. *Satellite-Terrestrial Integration Scenarios for Future Information-Centric Networks.*
- [6] Dirk Trossen, George Parisis, Borislava Gajic, Janne Riihijarvi, Paris Flegkas, Pasi Sarolahti, Petri Jokela, Xenofon Vasilakos, Christos Tsilopoulos, Somaya Arianfar, and Martin Reed. Architecture Definition, Components Descriptions and Requirements. FP7, oct 2011. Deliverable D2.3.
- [7] V. A. Siris, C. N. Ververidis, G. C. Polyzos, and K. P. Liolis. Information-centric networking (icn) architectures for integration of satellites into the future internet. In 2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL), pages 1–6, Oct 2012.
- [8] J. Lyu, Y. Chen, and Y. Cao. Ndn-based multimedia content distribution in space-ground integration network. In 2018 IEEE/CIC International Conference on Communications in China (ICCC Workshops), pages 69– 74, Aug 2018.
- [9] A. Thibaud, J. Fasson, F. Arnal, R. Sallantin, E. Dubois, and E. Chaput. An Analysis of NDN Congestion Control Challenges. In 2019 2nd International Conference on Hot Information-Centric Networking (HotICN), pages 18–24, 2019.
- [10] 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.
- [11] 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.
- [13] G. Carofiglio, M. Gallo, L. Muscariello, M. Papalini, and S. Wang. Optimal multipath congestion control and request forwarding in informationcentric networks. In 2013 21st IEEE International Conference on Network Protocols (ICNP), pages 1–10, Oct 2013.
- [14] Jim Griner, John Border, Markku Kojo, Zach D. Shelby, and Gabriel Montenegro. Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations. RFC 3135, June 2001.
- [15] S. Mastorakis and *al.* ndnSIM 2: An updated NDN simulator for NS-3. Technical Report NDN-0028, Revision 2, NDN, November 2016.