



HAL
open science

Analysis of the properties of NetcodICN protocols

Hirah Malik, Cédric Adjih, Michel Kieffer, Claudio Weidmann

► **To cite this version:**

Hirah Malik, Cédric Adjih, Michel Kieffer, Claudio Weidmann. Analysis of the properties of NetcodICN protocols. CORES 2020 - 5ème Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication, Sep 2020, Lyon, France. hal-02866093

HAL Id: hal-02866093

<https://hal.science/hal-02866093v1>

Submitted on 12 Jun 2020

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.

Analyse des propriétés des protocoles NetcodICN[†]

Hirah Malik^{1,2}, Cedric Adjih¹, Michel Kieffer², Claudio Weidmann³

¹Inria ²Univ. Paris-Saclay - CNRS - CentraleSupélec, L2S ³ETIS UMR8051, CY University, ENSEA, CNRS

Les réseaux centrés contenu (Information Centric Networking, ICN) sont une famille de protocoles récents basés un paradigme différent des réseaux IP classiques. Une extension naturelle et bien adaptée à ce type de réseaux est l'utilisation du codage réseau : ceci a été exploité dans la famille des protocoles NetcodCCN/NetcodNDN précédemment proposés, qui sont capables d'atteindre la capacité du réseau. Cependant, ceci a été observé avec des heuristiques, et sur quelques exemples de réseaux. Nous analysons plus formellement les propriétés de plusieurs variantes du protocole.

Mots-clefs : réseaux centrés contenu; information centric networking; ICN; codage réseau; network coding

1 Introduction

Information-Centric Networking (ICN) has recently emerged as an alternative to the traditional point to point communication [JSB⁺09]. The ICN principle is based on requesting and receiving data through names, by performing named based routing. IP addresses are not used. ICN removes the need to establish connections between the two endpoints and allows caching throughout the network.

An ICN network includes *consumer* applications that request content from *producer* applications by sending *Interest packets* carrying the content name, e.g. `/video/surprised-kitty/chunk/2431`. ICN allows consumers to use multiple interfaces/paths to receive the content. However, when multiple consumers request the same content, distributed over multiple producers in the network, coordination among the network nodes is required to efficiently use the network resources and avoid redundant content. As an alternative to solving this problem, *network coding* was introduced in [MWT12]. The main idea was to use *random linear coding* and send coded packets instead of original packets over the multiple available paths. One family of protocols, that we denote *NetcodICN*, combines network coding and ICN : NetcodCCN and NetcodNDN [SBTB16, Jon17] (on top of CCN and NDN [ASZ⁺18] variants of ICN respectively). They can reach capacity on examples, but one critical aspect is related to *interest forwarding algorithm*, that we study formally : mostly how parallel Interests reach the producer, on different variants.

2 Network Operation

The scenario in Fig. 1b summarizes ICN routing : consumer *A* sends Interest I_A that is propagated through the network by *routers*. When it reaches a producer that has the content that *satisfies* the interest, it is sent back in a *Data packet* (as **D**). ICN routing is different from classical routing : routers have a *Content Store* (CS) that cache named content (Fig. 1a). When a router receives an Interest that cannot be satisfied from its cache, it records the interest in a *Pending Interest Table* (PIT) as in Fig. 1a. Router makes the decision whether to *forward* the interest and on which faces. It records the face *f* from which the interest was received in *in-record* list (“*in : r_f*” in Fig. 1a), and on which faces it has been forwarded, in *out-record* (denoted “*out : s_f*” in figures). For simplicity of presentation, in the rest of this article, we will ignore the names as if there were a unique content with a unique name, as well as expiration issues.

Fig. 1c represents more precisely the PIT of routers, at the moment when the I_A reaches the producer. Each Interest also carries a random identifier, *nonce* : noted **A** for I_A . Router *X* recorded the reception of the Interest on face X_1 (“*in*”), and its forwarding on X_2 (“*out*”). Similarly for routers *Y* and *V*. The PIT entries globally record paths to the consumer *A*. Once the producer replies with a *Data packet*, it is propagated on one reverse path $Z \rightarrow Y \rightarrow X$. The alternate path $Z \rightarrow Y \rightarrow V \rightarrow X$ is ignored due to *nonce*. When receiving

[†]This research was [partially] supported by Labex DigiCosme (project ANR11LABEX0045DIGICOSME) operated by ANR as part of the program «Investissement d’Avenir» Idex ParisSaclay (ANR11IDEX000302)

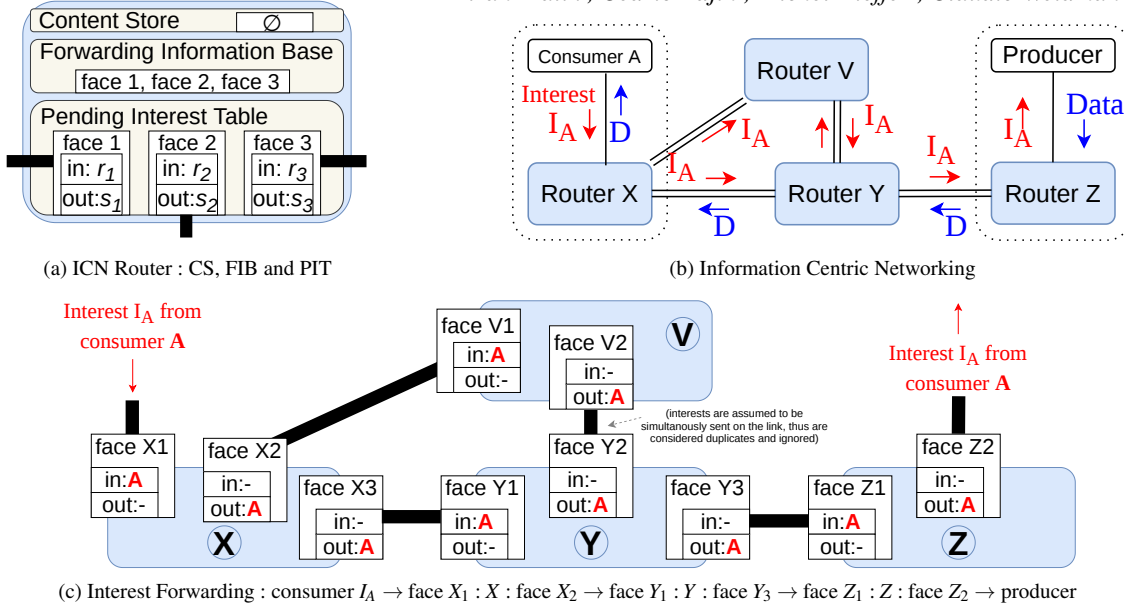


FIGURE 1: ICN principles

Data, the router check its PIT for pending interests for this data and forwards the data on the faces they were received (“in”). The PIT entries are finally removed. Routers also store the Data packet in their CS.

3 NetcodICN Semantics

NetcodICN is based on the idea that consumers send several undifferentiated Interests, in reply to each of which *linearly coded segments* (pieces) of the content will be sent, possibly from different producers. Receiving enough (linearly independent) coded segments will be sufficient to retrieve the original content. The core idea is that instead of keeping track of the *nonces* of the Interests in the PIT, a router can 1) allow for several identical interests from same face, 2) simply keep a *count* of them in the *in*-records and *out*-records (of Fig. 1a). NetcodICN is based on the following building blocks :

Information Bases : Each router maintains in the PIT, for each face $f : r_f =$ the counter of pending interests(the in-record) ; $s_f =$ the counter of the interests that have been forwarded on the face(the out-record) ; $\sigma_f =$ the number of coded segments that have been sent on the face ; $s_{all} =$ the total number of interests that required forwarding (from any face). The CS also stores some (linearly independent) coded segments $\{Q_1, Q_2, \dots, Q_\rho\}$. The number of these linearly independent segments defines the *router rank* ρ .

Forwarding Strategies : NetcodICN uses a Parallel Strategy (PS) that forwards Interests in parallel over all faces in the Forwarding Information Base (FIB), denoted \mathcal{F} . An interest is never sent back on its receiving face. The FIB \mathcal{F} can include all faces for instance, leading to a *Flooding* forwarding behavior denoted *FIB-F*, or only faces that can lead to a Producer without looping back to the node (denoted *FIB-P*) or the FIB of all routers could form a directed acyclic graph in that case the behavior is denoted *FIB-DAG*.

Interest Processing : when receiving an interest on a face f , the router first checks its rank, e.g. if it has full rank it could decode all coded segments. If so, the router can act a source and send a linear combination of packets. Otherwise, if $\sigma_f < \rho$; the interest can be immediately satisfied by sending one random linear combination of coded segments in the CS. Otherwise the interest is considered for forwarding.

Interest Forwarding : the router decides if it forwards the interest and where it forwards the interest. This is where we also identify variants (naming is ours) :

- **Optimistic Forwarding (OF)** : assumes that all sent interests on all faces will bring back content. Hence forwarding occurs when $r_f > \sum_{\phi \in \mathcal{F}} s_\phi$. This is proposed in [Jon17, p.53] (and NetcodNDN articles).
- **Pessimistic Forwarding (PF)** : counts the number of times s_{all} that the node had to do Interest forwarding. Forwarding is done when $r_f > s_{all}$. This is one interpretation of the proposition in [SBTB16][‡]

‡. There is a subtle ambiguity because [SBTB16, p.6] reads “For the sake of simplicity, we make the assumption that nodes follow a simple model in which any forwarded Interest brings an innovative segment before its expiration.”, but if an Interest is forwarded on three other faces, does it count as one “forwarded Interest” or three ? PF assumes one, OF assumes three.

- **Robust Forwarding (RF)** : the interest is forwarded on each interface $\phi \in \mathcal{F} \setminus \{f\}$ where less interests have been forwarded than received on f , e.g. $s_\phi < r_f$. For one single interest, this mimics NDN semantics [ASZ⁺18].

We assume consumers and producers do not forward interests, and that a consumer sends each of its interest on all its faces. Additionally, that counters are properly updated upon receiving Data (e.g. s_f, r_f, s_{all} , etc.).

4 Logical Analysis of the Behavior of NetcodICN protocols

The first important forwarding design trade-off depends on the FIB. FIB-F is straightforward to implement; FIB-P is often used instead of FIB-F[§], as it suppress obviously useless interests, but it requires knowledge of paths to producers; FIB-DAG requires a potentially expensive FIB computation protocol (e.g. NLSR [ASZ⁺18]) and some coordination of the nodes to avoid loops. Thus FIB-DAG is only suited for applications in smaller networks, but uses less network resources. We also note two important properties :

- **FIB-DAG is sometimes suboptimal** : in Fig. 2a, the use of link R_3-R_4 in both directions for consumers C_1 and C_2 is necessary to receive simultaneous content from P_1 and P_2 (assuming unit capacity links), creating a cycle, incompatible with a DAG. Hence FIB-DAG does not always allow maximum throughput.
- **FIB-P and FIB-F can generate overload** (with PF or RF) : content is quickly delivered to consumers, but as observed in our simulations, Data packets can be exchanged long after they get all content, because of replication of Interests. In Fig. 2b, if consumer C sends k Interests, it will get all content in $\frac{k}{2}$ units time, but router R_3 (and R_2) will receive content during k units time. This can reduce capacity for other content.

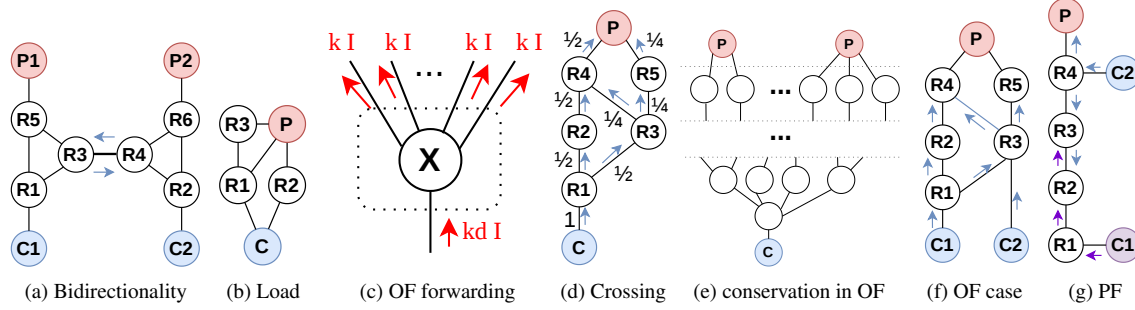


FIGURE 2

We now focus on the Interest forwarding algorithms : OF, RF, PF. One fundamental desirable property of such an algorithm is that : the number of Interests that reach the producer(s) is greater or equal to the number of Interests sent by the consumer(s). We denote this property *conservation*, for scenarios where no producers actually reply with Data packets (or with near-infinite delay).

OF Analysis, with one consumer : We first observe that OF has difficulties with conservation. This comes from the fact that a router of degree $d + 1$, when receiving kd Interests from one link, will send k interests on each forwarding face as in Fig. 2c. This is a form of *bounded* local flow conservation (output of one node is lower or equal to input). If Interest flow splits but meets again as in R_4 in Fig. 2d, some Interests will not be forwarded ($\frac{1}{4}$ in Fig. 2d), and total flow is not conserved. Thus cross links prevent conservation. With one consumer with one link (and any number of producers), conservation is possible with OF if and only if there are only parallel connections and no cross links, which implies a tree structure of the routers (before the producers) as in Fig. 2e. This is not even guaranteed by FIB-DAG.

OF with several consumers suffers from the same issues, except that the Interest flow from another consumer might compensate for flow losses for one consumer (Fig. 2f). Same situation can also occur for a single consumer with several outgoing links.

RF Analysis : RF will always provide conservation. This comes for its property that any node that receives k Interests on one of its link, will always forwards k Interests on each of its other faces in the FIB (either the k Interests, or some others

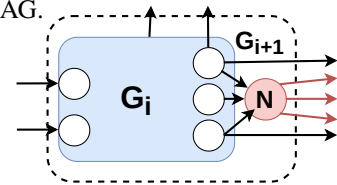


FIGURE 3

§. Not always clearly specify in all articles ; but default behavior of some simulators is equivalent to FIB-P

received previously). The sketch of a proof is in Fig. 3 : its represents if you have a graph G_i with that very property, you can expand it with any neighbor node to obtain a graph G_{i+1} with same property. E.g. k Interests incoming on the left, will still result into k Interests outgoing on the links of the new node N to the right (and vice-versa – not represented). Growing a graph iteratively (excluding consumers and producers), one can prove recursively that any k Interests sent by a consumer will result in k Interests reaching every router of the network (FIB-F), or at least every router on the path to producer(s) (FIB-P, FIB-DAG) and then every producer.

PF Analysis, with several consumers : We first observe, that in some scenarios, with FIB-F, PF will not provide conservation : in Fig. 2g, if Interests are sent simultaneously by consumers C_1 and C_2 , the k^{th} Interest of C_1 is blocked by the k^{th} Interest of C_2 at router R_3 because it has already forwarded the later.

PF with one Consumer : one can prove that with FIB-F, “if the (unique) consumer sends k Interests, then k Interests must reach every producer”[¶]. We denote this property \mathcal{P}_k . We first introduce a concept derived from a notable property from PF : let $\ell > 0$ be an integer, and imagine that a router u_i forwards its ℓ^{th} Interest (e.g. its s_{all} reaches ℓ). By definition, this is only possible if it has just received the ℓ^{th} of $r_f = \ell$ Interests on some face f . This face is linked to a previous router u_{i-1} ; we can similarly identify the previous hop u_{i-2} that caused u_{i-1} to forward its ℓ^{th} Interest, etc. and we establish a path, denoted ℓ -causality path, that starts from the consumer (no loop). With that definition, by induction on n , we prove the property \mathcal{P}_n : “ n Interests sent by the consumer will always result into n Interests reaching each producer”. \mathcal{P}_0 is obvious. We then assume \mathcal{P}_n is true, and prove \mathcal{P}_{n+1} by contradiction : we consider that $n + 1$ Interests are sent by the consumer. Notice \mathcal{P}_n can be applied to the first n Interests, hence at least n Interests will get forwarded by any router, thus either n or $n + 1$. Now denote U

the set of routers that forwarded exactly $n + 1$ Interests, and V the set of routers that forwarded n (Fig. 4). Assume that V is not empty, consider any router $w \in V$. Consider the n^{th} (and

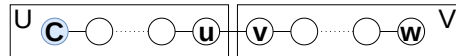


FIGURE 4

last) Interest forwarded by w , and the associated n -causality path of w . The n -causality path originates from the consumer c (in U), reaches $w \in V$, so it must cross the border between U and V : denote $u \in U$ and $v \in V$ such neighbor nodes on the n -causality path of w . By definition, the n^{th} forwarded Interest of u reaches v and is the n^{th} forwarded Interest by v . Now u will forward another $(n + 1)^{th}$ Interest. This Interest cannot originate from v , therefore it will be received by v . It will be the $(n + 1)^{th}$ received by v from u and v must forward another $(n + 1)^{th}$ Interest. This contradicts the fact that v is in V . Hence the hypothesis “ V is not empty” cannot be true, and the property \mathcal{P}_{n+1} is verified for $n + 1$, and by induction, for any n . \square

As a conclusion, this work has shown some important properties of NetcodICN with respect to Interest Forwarding. It illustrates that precision is required in the definition of the algorithms, and that their variants lead to different trade-offs and properties : some are not necessarily easy to establish nor known. We illustrate how logical analysis can be needed to establish some of them.

Références

- [ASZ⁺18] A Afanasyev, J Shi, B Zhang, L Zhang, I Moiseenko, Y Yu, W Shang, Y Huang, JP Abraham, and S DiBenedetto. NFD Developer’s Guide (rev. 10). Technical report, NDN-0021, 2018.
- [Jon17] Jonatan Eduardo Saltarin De Arco. *Network Coding Enabled Named Data Networking Architectures*. PhD thesis, University of Bern, 2017.
- [JSB⁺09] Van Jacobson, Diana K Smetters, Nicholas H Briggs, James D Thornton, Michael F Plass, Rebecca L Braynard, Nicholas H Briggs, and Rebecca L Braynard. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 1–12. ACM, 2009.
- [MWT12] Marie-Jose Montpetit, Cedric Westphal, and Dirk Trossen. Network Coding Meets Information-Centric Networking : An Architectural Case for Information Dispersion through Native Network Coding. NoM ’12, page 31–36, New York, NY, USA, 2012.
- [SBTB16] Jonatan Saltarin, Eirina Bourtsoulatze, Nikolaos Thomos, and Torsten Braun. NetCodCCN : A network coding approach for content-centric networks. *IEEE INFOCOM*, July, 2016.

¶. assuming consumer Interests have with sufficiently large spacing : when one arrives, the previous one has fully propagated