

Article

PIF and ReCiF: Efficient Interest-Packet Forwarding Mechanisms for Named-Data Wireless Mesh Networks

Dalbert Matos Mascarenhas ^{1,*} and Igor Monteiro Moraes ²

¹ Centro Federal de Educação Tecnológica Celso Suckow da Foseca—CEFET/RJ, Petrópolis, RJ 25620-003, Brazil

² Laboratório Mídiacom, PGC-TCC, Instituto de Computação, Universidade Federal Fluminense (UFF), Niterói, RJ 24210-346, Brazil; igor@ic.uff.br

* Correspondence: dalbert.mascarenhas@cefet-rj.br

Received: 31 August 2018; Accepted: 26 September 2018; Published: 29 September 2018



Abstract: In this paper, we propose three mechanisms to reduce the broadcast storm problem in wireless mesh networks based on the Named-Data Network (NDN) architecture. The goal of our mechanisms is to reduce the number of content requests forwarded by nodes and consequently, increase the network efficiency. The first proposed mechanism, called Probabilistic Interest Forwarding (PIF), randomly forwards content requests. The second mechanism, called Retransmission-Counter-based Forwarding (ReCiF), decides to forward content requests based on the number of retransmissions by adding a counter to the header of requests. The third mechanism, called ReCiF+PIF, combines the features of PIF and ReCiF to suppress content requests. We compare the performance of our mechanisms with both the NDN default forwarding mechanism and the Listen First Broadcast Later (LFBL) mechanism. Our proposals outperform the default NDN forwarding mechanism by up to 21% regarding the data delivery rate in dense networks and provide a 25% lower delivery delay than the default NDN. Our mechanisms accomplish this performance by only reducing the number of content requests forwarded by nodes. One of our mechanisms, PIF, outperforms LFBL regarding the data delivery rate and delivery delay by up to 263% and 55%, respectively, for high network contention levels.

Keywords: named-data networking; information-centric networking; NDN; wireless networks; broadcast storm

1. Introduction

Wireless Mesh Networks (WMNs) are envisioned to be a solution to provide low-cost access to the Internet [1]. Basically, WMNs employ a wireless backbone composed of stationary nodes to increase the connectivity compared with ad hoc networks. Client nodes, mobile or not, access the Internet through one or more than one of these backbone nodes because they are usually configured as gateways. For this reason, nodes that are often in the shortest paths toward gateways forward more packets than the others do and thus, consume more than their resources [2]. In addition to the high concentration of traffic near gateways, WMNs also suffer from interference and collisions that are characteristic of wireless transmission. Nodes try to access the medium at each hop because WMNs are based on multihop communication. Therefore, packets that already traversed several hops can be discarded near their destination, which leads to resource waste [3].

The information-centric networking (ICN) paradigm is a solution for the problems cited above in WMNs [4–7]. Information-Centric Wireless Mesh Networks focus on content delivery regardless of the content's physical location. Content requests and the contents themselves are forwarded based on names. Therefore, nodes have no addresses, and a content request is not necessarily addressed to a

particular node with a well-known address. The ICN paradigm also assumes that all network nodes can cache contents [8]. Thus, any node can send the content in response to a request if the desired content is stored in its cache. Consequently, with ICN, a content request does not necessarily reach the gateway to retrieve the content from the Internet if another node has previously requested the same content. In this case, the traffic load in nodes near to the gateway decreases as well as the number of hops traversed by the content.

Several architectures have been proposed in the literature to deploy the ICN paradigm both in the Internet and in wireless networks [9,10]. Named-Data Networking (NDN) [11] is one of these architectures. NDN radically changes the TCP/IP paradigm and becomes more suitable for handling the challenges faced by new network applications. Currently, users of applications, such as BitTorrent and NetFlix, are more interested in the content itself than in who provides this content [12]. In 2017, 52.88 % of the Internet traffic was generated by Netflix, YouTube, and BitTorrent applications [13]. The way that consumers request contents in NDN improves the data delivery rate when compared with other destination-oriented network architectures [11]. NDN nodes request content by sending an interest packet that is forwarded through the network until it reaches a node that is able to send back a data packet with the desired content. This node can be the content producer or a node that stores the content in its cache because, with NDN, intermediate nodes can cache contents. This approach reduces the bottleneck on a content producers path. That is the reason why NDN improves the data delivery rate. NDN also has native mobility support because consumers request contents based on names and not based on the destination address as in IP networks. Also, nodes do not change their addresses during changes of access points [14]. Another feature of NDN is the reduction of the control message overhead on the network. This reduction occurs because nodes do not need to propagate routing tables and link changes to their neighbors. These features classify NDN as a promising network architecture for the future networks, including wireless mesh networks [15].

Particularly for wireless mesh networks, in-network caching reduces the traffic concentration near gateways and the waste of resources caused by the late drop of packets. NDN-based WMNs can also benefit from the broadcast nature of the wireless medium to disseminate interest packets. Hence, when a node sends a packet, more than one neighbor is able to listen this packet transmission. Neighbors are thus candidates to forward a request or content. These multiple forwarding candidates increase the probability of a packet being delivered to producer nodes or to forwarding nodes which are temporarily able to provide contents from their cache.

The development of WMNs based on NDN, however, is a challenge because of the broadcast storm problem [16] caused by the forwarding of interest packets. With NDN, each node broadcasts an interest packet received for the first time. Thereafter, every neighbor will repeat the same behavior, leading to a broadcast storm of interest packets. In this context, nodes can experience a high collision rate depending on the number of consumers, i.e., nodes that request contents, and on the transmission rate of interest packets. Collisions increase packet losses and delay which are both caused by the retransmission of lost interest packets. This scenario characterizes the broadcast storm problem.

This paper proposes three mechanisms that limit the forwarding of interest packets in order to reduce the negative effects of the broadcast storm in WMNs. The goal of the proposed mechanisms is to decrease the number of interest packets forwarded by nodes and thus, to decrease the probability of collisions caused by these packets. Nodes, in this case, experience a higher delivery rate and lower delivery delay if the number of collisions decreases. The first proposed mechanism is called Probabilistic Interest Forwarding (PIF). With PIF, high-centrality nodes forward packets with a certain probability (p). With the second one, called Retransmission-Counter-based Interest Forwarding (ReCIF), nodes use the number of hops traversed by an interest packet in order to forward this packet or not. The third proposed mechanism, called ReCIF+PIF, combines the criteria of PIF and ReCIF. The idea is to guarantee that interest packets forwarded fewer times have priority to be forwarded by a node, and other interest packets are forwarded with a probability of p . Our three mechanisms are compared with both the default NDN forwarding mechanism and the Listen First Broadcast Later

(LFBL) mechanism. The content delivery rate, the delivery delay, and the overhead are the metrics considered in the simulation. The results show that the proposed mechanisms are more efficient in scenarios with a higher number of nodes contending to access the medium. For these scenarios, our mechanisms provide a content delivery rate that is 21% higher, with 25% less delivery delay than the default NDN forwarding mechanism. Our mechanisms also outperform LFBL regarding the data delivery rate and delivery delay by up to 263% and 55% respectively for high network contention levels.

The next sessions are organized as follows. Section 2 presents a brief overview of WMNs based on the NDN architecture. Section 3 discusses several works related to the adoption of caching and ICN architectures in WMNs. This section also presents proposals to deal with the broadcast storm problem in ad-hoc networks. Section 4 introduces the proposed mechanisms. Section 5 presents the simulation environment and discusses the simulation results. Section 6 presents the concluding remarks and comments about future work.

2. NDN-Based Wireless Mesh Networks

NDN employs two types of packets: interest and data packets. Interest packets are sent by nodes to request content. These nodes are called consumers. Data packets carry the requested content itself. These packets are sent by producers or by intermediate nodes that store the desired content in a cache. The storage of contents previously requested by an intermediate node in cache, called the Content Store (CS), creates a network of caches [17], as discussed in the next paragraphs.

Two data structures are used during the forwarding of interest packets in NDN. The Pending Interest Table (PIT) keeps track of the state of each transmitted interest packet that has not received an response yet, i.e., interests that are waiting for a data packet. Each PIT entry also records the receiving interface of the interest packet. The second data structure, called the Forwarding Interest Table (FIB), is used to forward interest packets to an output interface based on the content name. FIB maintains a list of entries containing name prefixes and output interfaces. The strategy layer of the NDN stack defines how FIB is populated.

The process of forwarding interest packets is described as follows. As soon as a node receives an interest packet, it verifies its CS in order to find a copy of the content requested. If the content is in its CS, the node sends a data packet towards the consumer through the receiving interface of the corresponding interest packet. Otherwise, the node verifies if there is another pending request for the same content on its PIT. If there is a PIT entry for the same content, the node updates the interface list of this entry by adding the arrival interface of the interest packet and dropping this interest packet. Otherwise, the node creates a new PIT entry with the arrival interface of the interest packet and looks up its FIB to determine the output interface to forward this interest packet. If there is no FIB entry related to the content name, the interest packet is dropped. Nodes repeat this forwarding process for each interest packet received. Data packets follow the reverse path traversed by interest packets because PIT stores the list of receiving interfaces of interest packets.

With NDN, intermediate nodes are able to cache contents previously requested by other nodes, which we call in-network caching. If employed by WMNs, in-network caching reduces the traffic concentration near gateways and the waste of resources caused by the late drop of packets, two of the main problems faced by this kind of network. Al-Arnaout et al. stated that the cost required to transfer data in wireless networks is higher than the cost required to transfer data in wired networks [4]. The reasons for this higher cost are the contention for the channel and channel interference. These two factors contribute to a reduction in network efficiency as the number of hops between the source and destination increases [18]. If intermediate nodes can cache contents, as NDN nodes do, this will reduce the number of hops between consumers and producers, consequently saving bandwidth by avoiding retransmissions caused by losses on the wireless channel. Oh et al. [5] argued that is better to invest in storage devices, which are getting cheaper by the day, than to deal with bandwidth limitations and losses of wireless networks.

Figure 1 illustrates an example of how contents are retrieved in NDN-based WMNs. The thin solid lines and thin dashed lines represent wired and wireless links, respectively. The solid arrows represent packets sent in wired networks, and the dashed arrows represent packets sent in wireless networks. Blue or heavy black arrows represent interest packets for the content (Y) sent by the first consumer (Step 1). Then, the data packet follows the reverse path towards the consumer. Each NDN router in this path stores a copy of content Y before forwarding the data packet and thus, the content availability increases. Thereafter, a new consumer requests the same content (Y) by sending interest packets represented by red or light grey, heavy-dashed arrows, Step 2. In this case, the interest packets do not reach the producer. Instead, this request is satisfied by an intermediate NDN node in the mesh backbone that has the content in its CS, as indicated. If nodes in the mesh backbone are not able to cache contents, all the consumers' requests for Y will be forwarded to the producer. In this example, in-network caching avoids bottlenecks near to producers, i.e., NDN reduces medium competition and traffic interference.

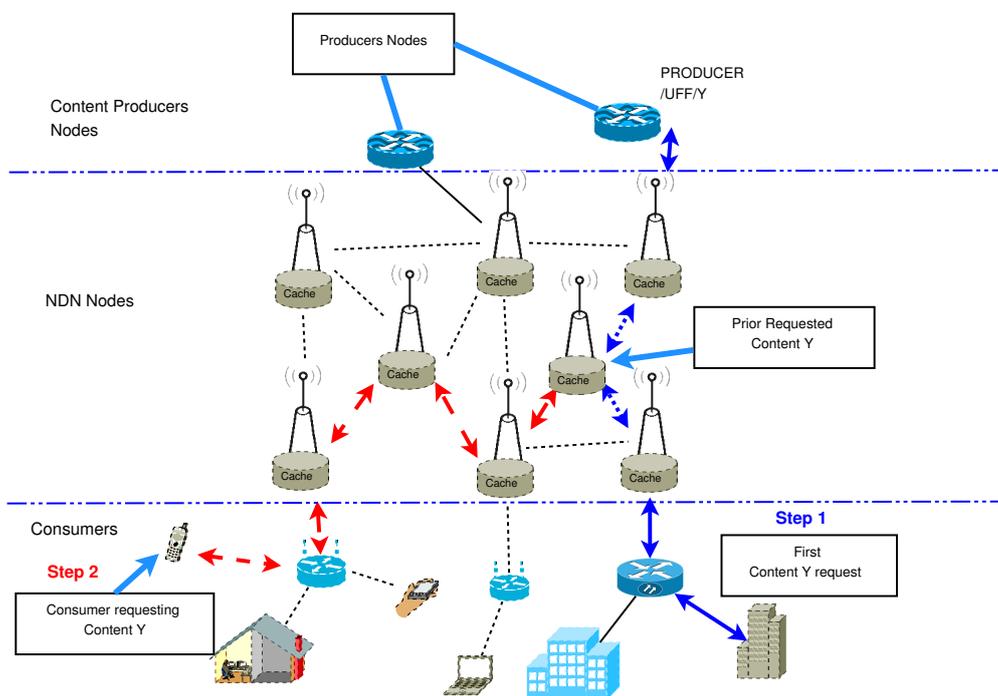


Figure 1. NDN-Based Wireless Mesh Networks.

There are two main storage policies proposed for NDN-based WMNs: reactive and proactive caching. Reactive caching policy evaluates whether received content will be cached or discarded based on the cache update. A simple way to use the reactive caching policy is to store contents only after an interest packet has been sent. On the other hand, proactive cache policies infer which contents should be predictably cached before requests arrive. In this case, nodes can cache a content even if there is no active PIT entry for this content. The use of a proactive cache in NDN-based WMNs can improve the average throughput, because nodes may take advantage of the broadcast nature of the wireless medium to store contents that have not yet been requested. Several proactive caching strategies use algorithms that predict which content should be cached based on consumers' demand profiles [18]. Liu et al. stated that the cache policy must take fading and interference on the wireless NDN into account [18]. These problems are shown when content is cached on distant neighbors instead of the nearest neighbor of the consumer. Therefore, the consumer receives the content from distant nodes where it is more prone to suffering interference from the nearest nodes along the path.

In addition to in-network caching, NDN-based WMNs can also benefit from the broadcast nature of the wireless medium to disseminate interest packets. Hence, when a node sends a packet, more than one neighbor can listen to this packet transmission. Neighbors are thus forwarding candidates which

forward a request or content. These multiple forwarding candidates increase the probability of a packet being delivered to the producer or to forwarding nodes, which are temporarily able to provide content from their caches.

Both the broadcast nature of the wireless medium and the way contents are requested, however, impose challenges to the deployment of NDN-based WMNs: the broadcast storm problem. With NDN, each node forwards an interest packet received for the first time. Then, every neighbor of this node that receives this interest will repeat the same procedure. We assume that nodes are often equipped with only one wireless interface. In this case, the interest packet is forwarded through the same reception interface, and all nodes in the transmission range of the forwarding node will receive the packet, even the node that previously forwarded this. Depending on the interest sending rate, nodes may experience a broadcast storm of interest packets. The consequences, in this case, are both an increasing medium access time and number of collisions. Collisions are caused by nodes that are in the same path used to obtain the content or by nodes that are closed enough to interfere each other [2].

Techniques have been proposed for wired NDNs to avoid the broadcast storm caused by interest packets. A simple technique is to define a rule per interface that states that interest packets must not be forwarded to the receiving interface. This technique, however, is not suitable for NDN-based wireless networks. There is no way to filter interest packets based on the forwarding interface because nodes are typically equipped with only one interface. In Sections 3 and 4, we discuss proposals to avoid the broadcast storm problem in WMNs.

NDN-based Wireless Mesh Networks are also suitable for reducing the deficiency faced by wireless mesh networks regarding mobility. That deficiency relies on most routing protocols developed for IEEE 802.11 ad-hoc networks [19]. Therefore, physical mobility can significantly affect the performance of those wireless mesh networks protocols. This occurs due to constantly route changing between endpoints [20]. Thus, those protocols require a level of track state for at least part of the topology and route configuration when links change. The reason for this is that highly dynamic networks impose different issues to IP addresses regarding the assignment and reassignment to the constantly changing set of active nodes in the network [21]. Depending on the application, these address modifications also need to be mapped to tables with names and IP address. Compared with NDN, that problem would be reduced since the requests are no longer destination-oriented and thus, are content-oriented.

Nowadays, mobile Internet traffic demand has increased the use of wireless networks, and the greater part of such traffic is content delivery, which is not real-time in nature [18,22]. This traffic is suitable for caching and inherits the possibility of content sharing between consumers. The cache use provokes challenges on the ambit of how the storage resource will be managed. It has been reported that, for both wired and wireless networks, caching consists of two main problems: content placement and content delivery [18]. For content placement issues, like cache size, the cache location and discarding policy are essential structures of the problem. The content delivery problem is related to the efficient use of network resources to deliver content to their consumers. Also, some researchers are conducting experiments to fulfill the need for a lookup service to scale network routing. The work of Afanasyev et al. [23] proposed a new Domain Name System (DNS) protocol for NDN, called NDN DNS (NDNS). The authors designed a DNS service to provide a lookup named data service without the need to announce name prefixes into the global routing system.

Another important aspect of NDN is the capability to use caches to promote popular contents. It has been reported that users send more than 300,000 tweets, share more than 680,000 pieces of content on Facebook, and upload 100 hours of video on YouTube per minute [24]. However few contents induce attention and concentrate most of the consumer requests [24]. Liu et al. [18] stated that the fundamental feature of content delivery traffic nowadays is that a minority of popular contents represent the majority of the traffic load and are consumed by different users at different periods. Based on this behavior, it is important to predict if and when content will become popular. Nevertheless, there is a lack of ability to predict the popularity of web content. This shortage is provoked by a sum of

aspects that impact directly on content popularity, for example, content quality and the importance of a content for consumers [25]. Those aspects are difficult to scope and may vary over time. Tatar et al. [24] argued that combining information from different media providers is the first step towards achieving better predictions. The authors also indicated that Twitter has been used for this accomplishment, but there are other promising content providers.

3. Related Works

Studies related to our work include (i) mechanisms that employ caches in WMNs and proposals to replace the TCP/IP architecture by NDN in wireless networks, (ii) the Listen First Broadcast Later (LFBL) protocol, and (iii) a summary of the related works. Discussions about these four topics are presented in the next sections.

3.1. Caching in Wireless Mesh Networks

Several authors have proposed the employment of caching in particular nodes of WMNs. MP-DNA (Multiple Partitions per Delegate Node Assignment) [4], for example, replicates contents to reduce the traffic overhead in WMNs. Thus, if a node needs to receive a retransmission of content or if any other node is interested in the same content, intermediate nodes can send it. MP-DNA runs on top of the TCP/IP stack and considers the Optimized Link State Routing Protocol (OLSR) to be running. Consequently, MP-DNA inherits the limitations of using TCP/IP in wireless environments. There are similar works to MP-DNA, such as MESHCHORD [26], that employ distributed hash tables to keep track of the content requested by neighbor nodes. In our work, we avoided the use of TCP/IP stack by adopting a stack that supports mobility natively.

Oh et al. [5] employed an NDN-based wireless network with mobile and stationary nodes. They compared both the NDN stack and the TCP/IP stack + OLSR in a military environment. Their results showed that nodes with NDN experience high delivery rates and low average delays. Gibson et al. [27] concluded that NDN provides advantages in highly dynamic and highly heterogeneous environments such as military coalitions. Regarding disaster management, NDN can also be used to disseminate alerts and to distribute countermeasure orders. Hannan et al. [28] proposed the detection of fire disasters by employing the NDN architecture in an IoT (Internet of Things) environment. NDN, in this case, helps to disseminate alert messages efficiently.

Kim et al. [9] proposed an NDN-based WMN packet delivery scheme using unicast transmission, called AIRDrop. To overcome the hidden terminal problem, the authors argued that AIRDrop uses the Request to Send/Clear to Send (RTS/CTS) mechanism to control link layer access in order to avoid collisions. Chatzimisios et al. [29], however, stated that RTS and CTS frames are sent at the basic link bit rate, which causes transmission delay. Another important fact presented in AIRDrop tests is that consumers only request different contents from each other. Consequently, the cache is partially used within the same message flow. Therefore, the gain related to prior content storage on intermediate nodes has not been well studied.

Amadeo et al. [30] proposed a content-centric mechanism to act upon the IEEE 802.11 link layer. This mechanism, called CHANET, uses the IEEE 802.11 broadcast feature to provide content-based routing services. The mechanism disrupts the concept of routing based on the destination address inherited from the TCP/IP protocol stack. According to the authors, NDN is suitable for ad-hoc wireless environments mainly because it improves the delivery rate in high mobility scenarios. The results show that CHANET increases the data delivery rate and also reduces the time required to deliver contents. Despite the fact that the authors only compared their mechanism with AODV (Ad hoc On-Demand Distance Vector Routing) protocol, they argued that representative experiments showed that NDN performs than OLSR.

Schneider et al. [31] proposed strategies to use multiple interfaces for mobile devices (LTE, Wifi, Ethernet) with NDN. The first strategy was to use the interface that best fits a specific application quality-of-service (QoS) requirement or even for a security requirement. The other strategy is to send

multiple copies of the same interest packet through different interfaces to guarantee redundancy. The objective is to access data regardless of the battery and throughput costs. Although the authors proposed strategies to use multiple interfaces, there is no strategy concerning broadcast storm control.

Lu et al. [32] proposed a social-tie based content retrieval (STCR) scheme used in delay-tolerant mobile ad hoc networks (MANETs). The authors used the K-means clustering algorithm to build the social level to forward messages. The work focused on sparse MANETs. In this scheme, all the nodes have an identifier and can compute their neighbor's ID and their centrality and social capacity to deliver a message to the destination or to forward the message to a better forwarding neighbor. The work differed from ours mainly because it was proposed for sparse networks, and the nodes exchange information about their social hierarchy before requesting content. Also, sparse scenarios are less prone to suffer from the broadcast storm problem.

Liu et al. [18] provided a deep study on strategies that use caches on base stations or end user devices instead of using the wired edge. The authors observed that the larger the cache size is, the higher the cache hit probability is and thus, the less the contention on backhaul bandwidth is. The authors also argued that in current cache literature, most of the studies have not take into account the channel fading and interference on wireless networks. Our work is focused on improving the throughput by reducing medium contention and interference.

Rainer et al. [33] proposed a testbed architecture providing a virtual wireless NDN over an IP network. The virtualization of NDN is modeled using applications such as Iptables and TC on Banana Pi Routers. In this scheme, all nodes receive a unique identifier to be used by forwarding strategies. The authors ran experiments both in the testbed and in the ndnSIM simulator and compared the results. The main conclusion was that virtualized NDN introduces a header overhead, but we consider this work to be a major step towards real NDN deployment.

NDN adoption has been also evaluated on cellular networks to reduce backhaul load and improve throughput. The idea is to cache popular contents on base stations (BSes) and in some cases, mobile devices. Zhao et al. [34] proposed a framework that reduces the complexity of cellular networks and increases network throughput by using NDN. This work also showed that caching popular contents is well desired for user satisfaction. The authors also showed that as a consequence of using a content-centric approach, the interference among the small cell networks was reduced and thus, the cost of the implementation is also reduced.

3.2. Listen First Broadcast Later

The Listen First Broadcast Later (LFBL) protocol [20] is classified as a content-oriented routing protocol, and there are several studies that mention it in the literature. This is the reason why we compared our mechanisms with LFBL and the default NDN in Section 5.

With LFBL, routing decisions are made by the node that receives a packet and not by the node that sends the packet. Therefore, after receiving a packet, a node considered to be a potential router listens to the channel for a while to verify whether there are other nodes more suitable than the node itself to be the packet forwarder. During the listening time interval, if there is no better candidate to forward the packet, the node itself will forward the packet. This procedure is called the request phase.

LFBL does not maintain a state of connections. Therefore, nodes that are in the path of an end-to-end flow evaluate whether they are capable of being forwarders based on their distances to the destination. This behavior removes the need for knowledge of network topology from LFBL. By using broadcast communications, nodes that are in the listening perimeter can decide if they can forward the packet. Hence, all the information related to routing is within the packet, as LFBL only employs broadcast communications for sending packets.

The listening period is a key parameter of the LFBL protocol. Assuming that in a wireless mesh network there is more than one node transmitting its data, the listening time for a given node should be long enough to receive the neighbor packets and verify if it is a forwarder or not. The authors argue that this time serves primarily to give priority to communications and also to avoid collisions. Priority

refers to giving shorter time to nodes which have already chosen candidates for forwarding the packets. In this way, nodes that have previously forwarded packets from a particular flow have priority over nodes that have not yet forwarded packets from this flow. The listening time is also used to reduce the collision because two forwarding nodes do not listen during the same period. By listening for a period, two nodes can consequently send the data precisely at the same time, as both act as if they are the best candidate to forward the packet. Therefore, the authors argue that using a random factor for the listening period considerably improves the protocol's performance.

LFBL creates different flows between the source and the destination. These flows, after the initial request phase, are destination-oriented. Hence, for a densely populated network with a high degree of requests, these flows can create bottlenecks in the neighborhood of producers. We have also observed that if multiple content servers are in the network, the calculation of the producer distance becomes imprecise. This imprecision is because for different responses of different content servers, the intermediate nodes will assume their eligibility in a precise way that does not necessarily reflect the best routes.

The description of the LFBL does not detail how caching is done. For the sake of fairness, we enabled the LFBL cache with the same size used by our mechanisms. LFBL also does not detail how long nodes have to wait during the request flooding phase. In our analysis, we observed that this time period severely impacts LFBL performance. For our simulations, we used 20 s as the waiting time.

3.3. Summary

Besides the related works that aim to reduce the broadcast storm problem, there are remaining challenges concerning the strategies to reduce medium contention by interest packets. These problems are related to interest packet suppression combined with the number of nodes in the network. Therefore, it is necessary to choose the right criteria for suppressing interests so as to not reduce the network efficiency. Some related works present strategies to reduce the broadcast storm from interest packets, i.e., LFBL, but when more producers are introduced or the network contention is increased, the performance of these mechanisms decreases. There are related works that combine IP networks and NDN. Hence, these strategies are based on adaptations to use NDN over the IP layer, therefore inheriting TCP/IP related problems. Our mechanisms present robustness concerning network contention and the number of producers. Hence, our mechanisms do not require TCP/IP overlay, thus providing entirely information-centric forwarding strategies.

4. The Proposed Mechanisms

We propose three mechanisms to reduce the broadcast storm problem in NDN-based WMNs [35]. The goal of our mechanisms is to reduce the number of interest packets forwarded by nodes and consequently, to increase network efficiency regarding the delivery rate and delay. The following sections introduce the proposed mechanisms and discuss implementation details.

4.1. The PIF Mechanism

The first proposed mechanism is called Probabilistic Interest Forwarding (PIF). With PIF (Algorithm 1), a node forwards interest packets with a probability of p if the node has a centrality degree ($g \geq G$), where G is the centrality threshold. Otherwise, nodes that have a centrality degree ($g < G$) forward interest packets according to the default NDN forwarding mechanism. The interest packet forwarding probability (p) is the same for all nodes. Each node has a centrality degree (g) that refers to the number of neighbors that a node has at the time of the interest packet forwarding decision. The centrality threshold (G) is defined based on the network density. It stands for the number of neighbors considered to decide whether the node has high centrality or not, and it is the same for all nodes.

With PIF, we aim to reduce the number of interest packets sent by nodes with a high degree of centrality, which means nodes with a high number of neighbors. Preliminary experiments suggest the definition of a forwarding criterion based on the centrality degree. We observed more nodes contending for the medium just after an interest packet forwarding by a node with more neighbors.

These nodes are trying to send copies of the same interest packet. Besides, a neighbor of a node with a high degree of centrality has a high probability of having already received that interest from another node, because interest packets are sent in a broadcast.

Algorithm 1: Probabilistic Interest Forwarding (PIF) Mechanism.

```

Data: (p, g, G, pck.interest.)
1 begin
2   if g >= G then
3     λ ← Random();
4     if λ ≤ p then
5       Forward(pck.interest)
6     else
7       Discard(pck.interest);
8     end
9   else
10    Forward(pck.interest)
11  end
12 end
    
```

Figure 2 presents a scenario where a consumer node requests content from a given producer. At time $t = 1$, node X has already received the interest packet sent by the consumer and then broadcasts it. After this, at time $t = 2$, nodes C and F receive the interest and then broadcast it to their neighbors. Let us suppose that A is a central node and decides to not forward the interest packet based on the probability (p) defined by PIF. In this case, we have less contention for the medium. Otherwise, if A forwards this interest, C, F, E, D and B can be impacted. In this example, we observe that suppressing some interest packets in central nodes reduces the medium contention. Therefore, we increase the forwarding probability of other interest packets or contents in the transmission channel and thus, increase the network throughput.

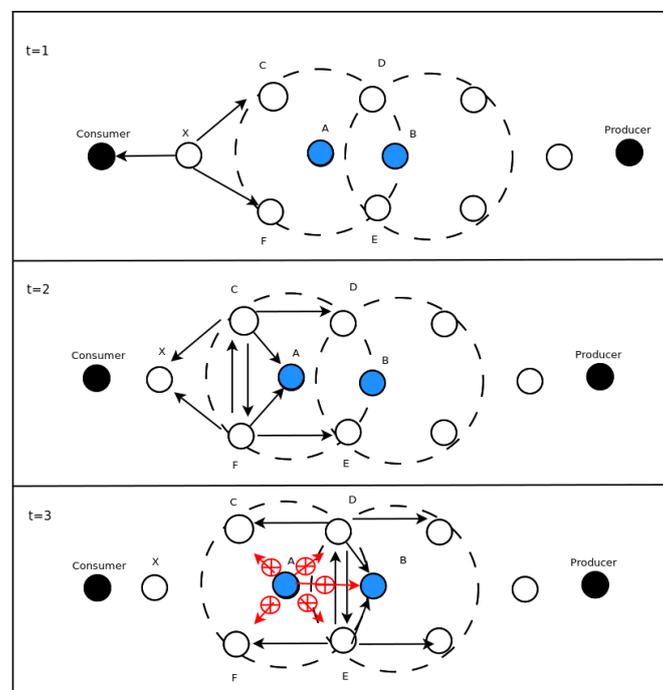


Figure 2. An example of the PIF mechanism in action.

4.2. The ReCIF Mechanism

The Retransmission-Counter-based Interest Forwarding (ReCIF) mechanism introduces one field in the interest packet header (Algorithm 2). This field, referred to as a forwarding counter, receives zero

as initial value and is increased by one unit each time that the packet is forwarded. Nodes that have a degree centrality ($g \geq G$), where G is the centrality threshold, forward an interest packet only if $c_i < C$, where c_i is the value of the forwarding counter in the header of packet i , and C is the retransmission threshold. Otherwise, nodes that have $g < G$ forward interest packets based on the default NDN forwarding mechanism. ReCIF assumes that the forwarding counter of a packet increases quickly during broadcast storm periods. Thus, ReCIF limits the forwarding of packets that were already in a broadcast storm. The fewer the number of copies of these packets, the lower the collision probability is.

Algorithm 2: Retransmission-Counter-based Interest Forwarding (ReCIF)-hard mechanism.

```

Data: ( $g, G, C, c_i, pck.interest.$ )
1 begin
2   if  $g \geq G$  then
3     if  $c_i \leq C$  then
4        $Forward(pck.interest);$ 
5     else
6        $Discard(pck.interest);$ 
7     end
8   else
9      $Forward(pck.interest)$ 
10  end
11 end

```

Defining the retransmission threshold (C) is a challenge in the employment of ReCIF. If C is low, interest packets may not reach their destinations. Otherwise, if C is high, the efficiency of ReCIF is compromised. Therefore, we propose two operation modes for ReCIF: ReCIF-hard and ReCIF-soft. With ReCIF-hard, C is fixed and has been previously defined based on the network density. The threshold C is the same for all contents. On the other hand, ReCIF-soft calculates C dynamically (Algorithm 3). Thus, with ReCIF-soft, each content (n) has its own threshold (C_n).

Algorithm 3: ReCIF-soft mechanism.

```

Data: ( $g, G, c_i, pck.interest.$ )
1 begin
2   if  $g \geq G$  then
3     if then
4        $ReceivedPcksN \leq P$ 
5     else
6        $sumCi \leftarrow c_i + sumCi;$ 
7       if then
8          $ReceivedPcksN == P$ 
9       end
10       $C_n(t) \leftarrow sumCi / P$   $Forward(pck.interest);$ 
11     end
12      $C_n(t) \leftarrow ((1 - \alpha) * c_i + \alpha * C_j(t - 1));$  if  $c_i \leq C_n(t)$  then
13        $Forward(pck.interest);$ 
14     else
15        $Discard(pck.interest);$ 
16     end
17   else
18      $Forward(pck.interest)$ 
19   end
20 end

```

ReCIF-soft employs a two-phase algorithm to calculate threshold C_n . The first phase occurs when a node receives an interest packet for a specific content for the first time. To calculate the first value of C_n , a given node has to receive P interest packets. All of these P interest packets are forwarded by

the node following the default NDN forwarding mechanism. After that, the node calculates the mean value of the forwarding counter (c) of these P packets and sets $C_n(0)$, as defined in Equation (1),

$$C_n(0) = \left(\sum_{i=1}^P c_i \right) / P. \tag{1}$$

We defined $P = 10$ based on preliminary experiments.

With $C_n(0)$ defined, a node starts the second phase of the algorithm. To calculate C_n dynamically, ReCIF-soft employs an exponential moving average defined by Equation (2):

$$C_n(t) = (((1 - \alpha) * c_i + \alpha * C_n(t - 1))), t > 0. \tag{2}$$

The idea is to prioritize prior values of interest retransmissions and give small weights to the recent ones. We used, by default, $\alpha = 0.9$ to adjust the threshold C_n for every interest received for the same content. This small weight can modify the threshold on a low scale. Therefore, high hop count values from few packets will produce a reduced impact on C_n .

If a received interest packet of content n has c_i less or equal to C_n , then this packet will be forwarded, and the new C_n value will be computed. Otherwise, if c_i is higher than C_n , the packet will be dropped, but the new C_n will be computed. The reason for computing C_n from a discarded interest packet is to maintain the protocol awareness upon new network changes.

Figure 3 presents two scenarios where a central node B will decide if it forwards or discards a received packet. The blue or grey nodes are the central nodes and are running the ReCIF-soft mechanism. Let us assume that the second scenario initiates after the first scenario. For the first scenario, we assume that $C_n(t) = 5$, due to prior calculations. Thus, when the packet arrives at node B , after being forwarded four times, its c_i is 4. The node then compares c_i and $C_n(t)$ and then accepts the forwarding of the packet, because the packet was forwarded less than five times. After this, B calculates the new value for $C_n(t)$ which is $(4 * 0.1 + 5 * 0.9) = 4.9$. Thereafter, for the second scenario, the value of $C_n(t)$ is 4.9. The packet received by B in the second scenario has been forwarded seven times before, which is higher than $C_n(t)$. Therefore, the packet will be discarded, and the new $C_n(t)$ will be recalculated to 5.11.

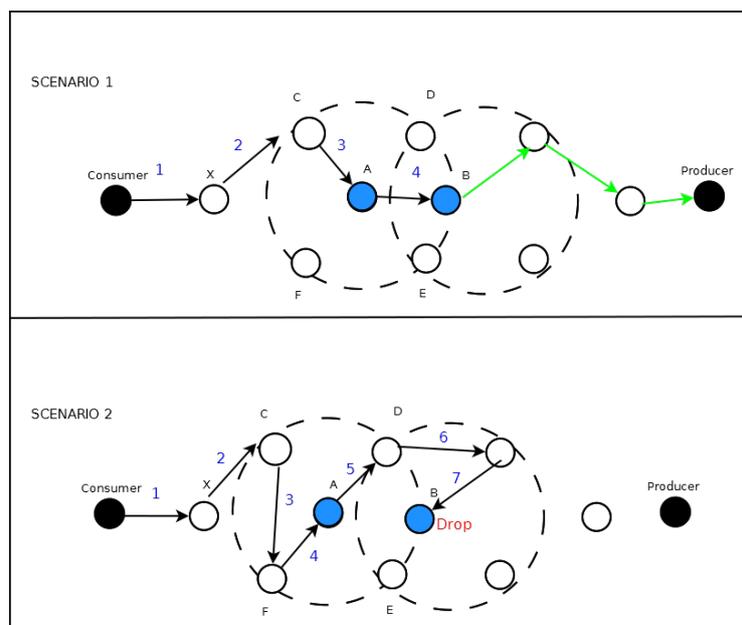


Figure 3. An example of ReCIF-soft in action.

We also evaluated ReCIF-soft with the threshold $G = 0$ (Algorithm 4). In this scenario, all nodes forward interest based on the criterion adopted by ReCIF-soft, regardless of its centrality degree. The idea

behind this strategy is to avoid the broadcast storm, even on nodes located in low-density regions of the network. In some experimental traces, we observed that nodes with a low centrality degree also contribute to increasing the amount of interest forwarded by nodes with a high degree of centrality.

Algorithm 4: ReCIF-soft $G=0$ Mechanism.

```

Data: ( $c_i, pck.interest.$ )
1 begin
2   if then
3     |  $ReceivedPcksN \leq P$ 
4   else
5     |  $sumCi \leftarrow c_i + sumCi$ ;
6     | if then
7       | |  $ReceivedPcksN == P$ 
8     | end
9     |  $C_n(t) \leftarrow sumCi/P$   $Forward(pck.interest)$ ;
10  end
11   $C_n(t) \leftarrow ((1 - \alpha) * c_i + \alpha * C_j(t - 1))$ ; if  $c_i \leq C_n(t)$  then
12    |  $Forward(pck.interest)$ ;
13  else
14    |  $Discard(pck.interest)$ ;
15  end
16 end

```

4.3. The ReCIF+PIF Mechanism

The third proposed mechanism (Algorithm 5), called ReCIF+PIF, combines the features of PIF and ReCIF-hard. ReCIF+PIF also introduces the forwarding counter in the header of interest packets. Similarly, the counter receives zero as initial value and is increased by one unit each time the packet is forwarded. Therefore, nodes with degree centrality ($g \geq G$), where G is the centrality threshold, forward interest packets only if $c_i < C$, where c_i is the current value of the forwarding counter in the interest packet (i), and C is the retransmission threshold. On the other hand, nodes that have degree centrality ($g \geq G$) forward interest packets that have $c_i > C$ with probability p . Thus, with ReCIF+PIF, it is less critical to determine how to define C because packets that are classified above this threshold still have a probability (p) to be forwarded. Nodes that have degree centrality ($g < G$) forward interest packets according to the default NDN forwarding mechanism.

Algorithm 5: ReCIF+PIF mechanism.

```

Data: ( $p, g, G, C, c_i, pck.interest.$ )
1 begin
2   if  $g \geq G$  then
3     | if  $c_i \leq C$  then
4       | |  $Forward(pck.interest)$ ;
5     | else
6       | |  $\lambda \leftarrow Random()$ ;
7       | | if  $\lambda \leq p$  then
8         | | |  $Forward(pck.interest)$ 
9       | | else
10      | | |  $Discard(pck.interest)$ ;
11      | | end
12      | | ;
13    | end
14  else
15    |  $Forward(pck.interest)$ 
16  end
17 end

```

4.4. Implementation Details

With ReCIF and ReCIF+PIF, nodes need to know the number of hops traversed by a packet. The nodes running our three mechanisms also need to be aware of their centrality degree. For these purposes, we added two fields to the interest packet header: FWD_counter and node_ID. Table 1 summarizes the fields added to the interest packet header.

Table 1. New fields added to the interest packet header.

Interest Packet Field	Description
FWD_counter	Incremented at each packet forwarding.
node_ID	Receives the identifier of the last forwarding node, modified at each hop.

Each time the packet is forwarded, the FWD_counter field is incremented by one. The purpose of this field is to compute how many retransmissions a given packet has passed through, i.e., the number of hops traversed by the packet. There have been several proposals for neighbor discovery mechanisms in wireless networks [36,37]. For simplicity, we added a field to the interest packet header for this purpose. The node_ID receives the identifier of the last node that forwarded the interest packet. The node only puts its identification when it forwards the interest. Therefore the other nodes, upon receiving the interest, can count the different node_ID to obtain an estimate of neighboring nodes. The node_ID field is only used for neighbor discovery, not for forwarding purposes. Therefore, we do not violate the NDN paradigm.

Figure 4 explains how a node calculates its centrality threshold:

1. After the arrival of the interest packet, the node checks its neighbor list size.
2. If the list of neighbors, called *neighbor.LIST*, is empty, the node adds the tuple node_ID, obtained from the interest packet header, and the interest packet arrival time, *pckt.TIME* to the list. After that, the neighbors list size *neighbor.LIST.Size* is incremented.
3. If the list of neighbors is not empty, the node ID extracted from the packet is compared with IDs stored on the *neighbor.LIST*. If the ID is not found in the neighbor list, then the ID and the packet arrival time are added to the list, as previously described. Otherwise, if the ID is found on the list, then the packet arrival time is updated (*pckt.TIME*). A timeout is defined for each entry. If the timeout expires, the tuple is removed from the list, and the list size is updated. The timeout is used to remove nodes from the list that may be off because of some failure or administrator change. The amount of time used to set the expiration time might change to cover the specific network flexibility. For our work, that time was set to the time used in the simulation, since our nodes were fixed and the environment was controlled regarding node failures.
4. The last step is to check if any tuple of the neighbor list has a packet time higher than the expiration time. In such cases, the tuple will be removed, and the neighbor list size will be decremented.

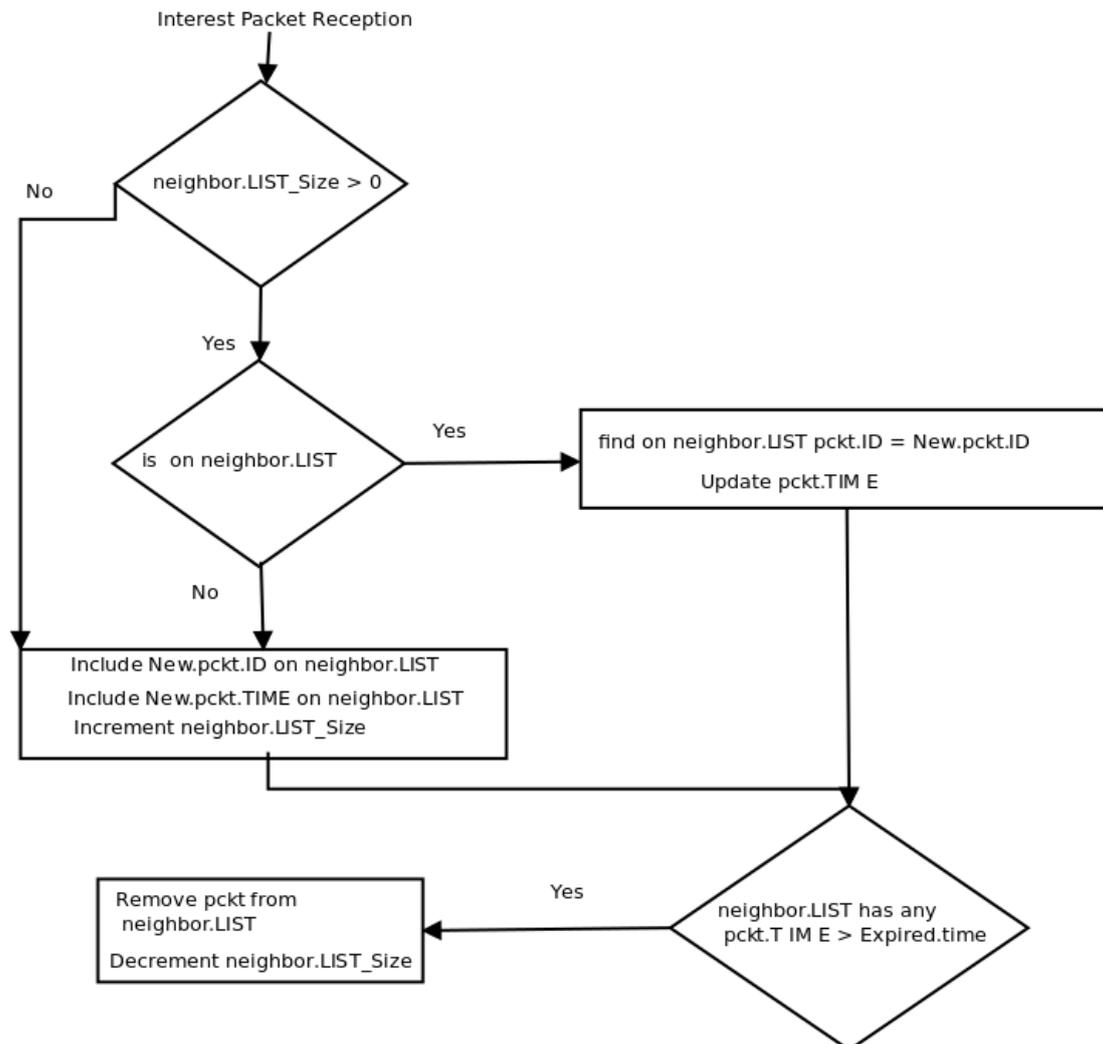


Figure 4. Neighbors discover flowchart.

5. Results

We evaluated the proposed mechanisms through simulation by using the NDNsim module [38] of the NS-3 simulator. NDNsim implements the default NDN architecture. We modified this module to add support for wireless communications and our proposed mechanisms. In the following sections, we present the simulation parameters used in our evaluation. First, we conducted experiments to investigate the performance of NDN when compared to the TCP/IP stack running the OLSR protocol on a WMN-based scenario. We also conducted experiments to parametrize our mechanisms and then we compared the performances of PIF, ReCIF, and ReCIF+PIF with both the default NDN forwarding mechanism and LFBL.

5.1. Simulation Parameters

The simulation considered a real network topology running in an urban scenario. We used real traces from the CRAWDA [39] project that were acquired from a wireless mesh network operating in Portland, Oregon, USA, depicted by Figure 5. This network covers an outdoor scenario with urban buildings between some nodes. The network consists of 70 nodes, and some of them are geographically separated by a river. The reason for choosing this network was related to the number of nodes and their positions. Nodes are densely concentrated in some areas while in other areas, nodes are barely sparse. Such different densities provided an opportunity to evaluate our mechanism in a realistic scenario. Producer nodes were fixed on the edge of the network, while the consumer nodes were

randomly distributed. To simulate this scenario, which is urban and outdoor, the following parameters for the link and physical layers were defined. The network technology was the IEEE 802.11a standard in ad-hoc mode with OFDM (Orthogonal Frequency-Division Multiplexing) modulation and the transmission rate was 24 Mb/s. The propagation model used was the Shadowing Deviation with 5 dBm of transmission power.

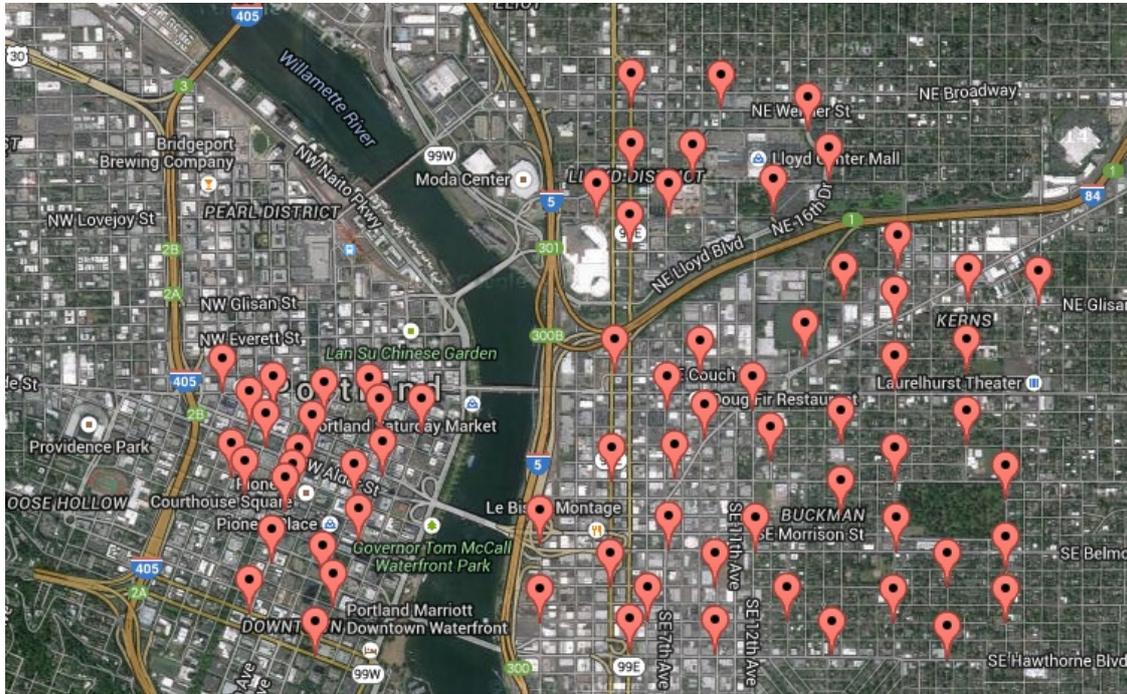


Figure 5. Geographical location of Portland network nodes.

We varied the transmission rate of interest packets. Consumers sent from 10 to 50 interest packets per second. The interest packets carried the required content names. Every item of content had a size of 1000 bytes and was encapsulated in one data packet. The cache size was equal to 10,000 kB in all nodes and the cache replacement policy was the Least Recently Used (LRU). Depending on the scenario analyzed, content chunks were requested sequentially or based on their popularity levels, defined by a Zipf-like distribution with the parameter $\alpha = 0.7$.

We defined the following parameters for the proposed mechanisms. Both PIF and ReCIF+PIF were evaluated for $p = 0.2$. The reason for this value is explained in Section 5.3.1. The centrality threshold (G) was equal to 8. Based on preliminary experiments, this is the average number of neighbors of nodes that are placed in high-density areas of Portland Network. Finally, with ReCIF-hard, the retransmission threshold used was $C = 7$. This value was also defined according to preliminary tests to determine the average number of hops traversed by interest packets sent by a given consumer to reach the producers in the Portland Network. ReCIF-Soft used $\alpha = 0.9$ and $P = 10$.

The metrics considered in the simulation were the number of content packets delivered, the average delivery delay, and the number of interest transmitted by contents received. The rate of content packets delivered was calculated using the number of requested contents delivered on each consumer node. The content delivery delay is the time between the transmission of an interest packet and the reception of the corresponding data packet by a consumer. For each configuration of our mechanisms, described in Section 4, we performed 50 simulation runs of 100 s each. Confidence intervals were calculated at the 95% confidence level and were represented by vertical bars in figures in the following sections.

5.2. TCP/IP Stack with OLSR vs NDN

First, we compared the WMN based on the default NDN with a WMN based on the TCP/IP stack running the OLSR protocol. The main objective of this experiment was to investigate whether NDN outperforms the TCP/IP stack in mesh networks. The parameters of OLSR used were the following. The interval between HELLO messages was equal to 1.5 s. Link state messages were sent every 5 s. These messages were used by multipoint relays to calculate the topology map. We assumed traffic with a Constant Bit Rate (CBR) to simulate the request and the delivery of contents in the TCP/IP + OLSR WMN. CBR packets had a size equal to NDN data packets, i.e., 1000 bytes. The goal of this configuration was to approximate the traffic behavior of the two networks, because both networks operate in very different ways regarding data forwarding. Therefore, the CBR traffic was transmitted at the same rate as contents were requested by NDN consumers.

The TCP/IP + OLSR network is not able to cache popular contents; therefore, we considered contents to be requested sequentially for the sake of a fair comparison. Figure 6 shows the results for the content delivery rate and average delay. We varied the number of consumers between 2 and 30, and varied the rate that each consumer sent content requests between 10 and 50 interests per second. The idea was to investigate the impact of the number of consumers and the rate of content requests in the two WMN deployment approaches. In Figure 6a, the vertical bars represent the default NDN and TCP/IP + OLSR network for 2 or 30 consumers. Bars are grouped based on the rate of interest sent by consumers, 10 or 50 int/s. The results show that the default NDN delivered more content packets than TCP/IP + OLSR for all analyzed configurations. Furthermore, the performance gap between the two approaches increased as both the number of consumers and the rate of content requests increased. For example, NDN delivered almost 100% of the content packets for two consumers, sending 10 int/s. TCP/IP + OLSR delivered 70% in this configuration. When the number of consumers increased to 30, however, NDN delivered almost 90% of the content packets and TCP/IP + OLSR delivered only 30%.

We also observed that the delivery rate severely decreased if the content request rate increased to 50 int/s. In this case, the network was saturated, and as a consequence, more packets were lost by medium contention and inter-flow interference. The default NDN performance was still higher than the one provided by TCP/IP + OLSR. This performance is a consequence of proactive caching employed by the default NDN. With proactive caching, nodes store contents sent by other nodes in their transmission range, even if there is no PIT entry for the content. Thus, the content availability increases. With NDN, consumers retrieve contents from adjacent nodes and consequently interest and data packets are forwarded fewer times. Thus, the medium contention is reduced. On the other hand, with TCP/IP + OLSR, packets always traverse the source-destination path.

In Figure 6b, it can be observed that the default NDN provided a higher average delay than TCP/IP + OLSR with 30 consumers. In addition, the higher the network saturation was, the higher the average delay was for both approaches. The reason for this behavior is because without the use of a cache, remote nodes on the TCP/IP + OLSR tend to receive fewer packets than NDN. By computing the reception of the contents that have traversed more hops, the average delay is also increased.

Thus, we conclude that NDN delivers more contents than TCP/IP + OLSR. This increase on content delivery rate, however, has a drawback related to a higher delivery delay. Network saturation is the key point to reduce the content delivery rate and increase the content delivery delay. Therefore, we propose mechanisms to reduce the number of interest packets transmitted and thus, to reduce the network saturation.

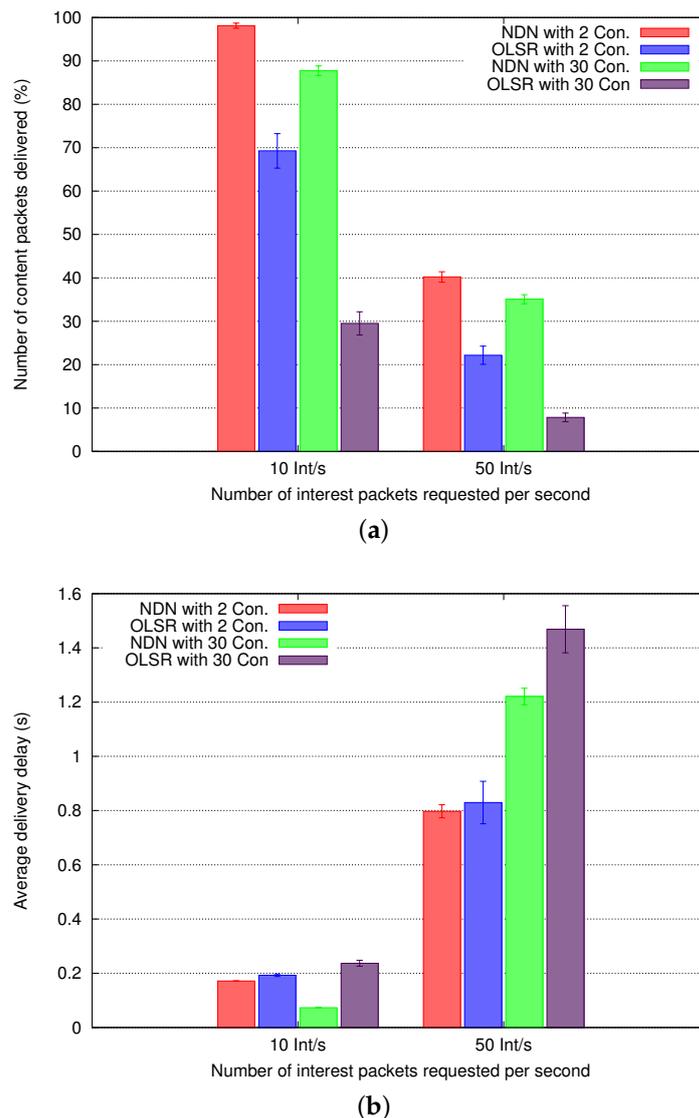


Figure 6. Default NDN vs. TCP/IP with OLSR: content delivery rate and average delay. (a) The percentage of content packets delivered by the TCP/IP stack with the Optimized Link State Routing Protocol (OLSR) and by default Named-Data Networking (NDN). (b) The average delay of content packets delivered by TCP/IP stack with OLSR and by default NDN.

5.3. The Proposed Mechanisms

We compared the performances of the proposed mechanisms, PIF, ReCIF, and ReCIF+PIF, with both the default NDN forwarding mechanism and LFBL. We considered three metrics to evaluate the mechanisms: the content delivery ratio, the average delay of delivered contents, and the overhead. The first two metrics were explained previously, and overhead stands for the number of interest packets forwarded for successfully received contents. Each consumer requested content based on popularity according to a Zipf-like distribution. We varied the content request rate, the number of consumers, the cache size, and the network size. The following sections discuss the results.

5.3.1. Parametrization

We first conducted experiments to define the interest packet forwarding probability (p) used by PIF and ReCIF+PIF. We varied this probability from $p = 0.1$ to $p = 0.9$ considering 30 consumers sending interest at a rate of 50 interest/s. Figure 7 shows the results. We observed that the number

of content packets delivered was the highest when $p = 0.2$ or $p = 0.3$. On the other hand, PIF with $p = 0.2$ provided a lower average delivery delay and overhead than PIF with $p = 0.3$. Thus, we defined PIF nd ReCIF+PIF as having $p = 0.2$ for the experiments presented in the next sections.

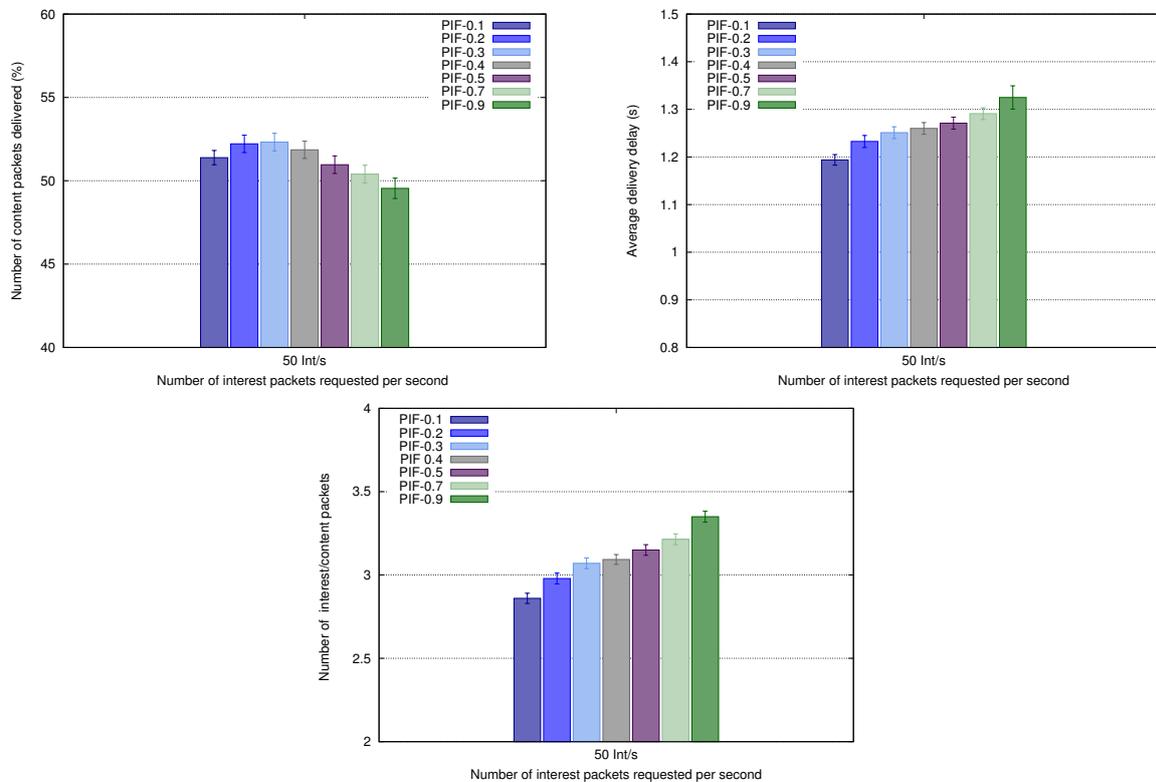


Figure 7. Content packets delivered, average delivery delay, and overhead for 30 consumers requesting popular contents with PIF considering different interest forwarding probabilities.

5.3.2. The Impact of the Content Request Rate

We compared our mechanisms with the standard NDN and LFBL in a scenario with a high number of consumers with different content request rates. We considered 30 consumers randomly chosen at each simulation run, requesting contents available in two producers, and we varied the content request rate from 10 to 50 int/s.

Figure 8 compares the default NDN, LFBL, and our mechanisms. First, we conclude that the proposed mechanisms outperformed the default NDN and LFBL regarding the number of content packets delivered. Less restrictive mechanisms, such as ReCIF and ReCIF+PIF with $p = 0.2$ (RECIF02 for the sake of brevity), provided higher delivery rates. With an increase in the content request rate, the nearest nodes of producers were more likely to receive contents. In the more saturated scenario, with 50 int/s, the difference from LFBL to ReCIF, regarding content delivery, 263.4% more contents used ReCIF. The reason for this behavior lies in the fact that LFBL uses multiple independent flows to request contents. Therefore, LFBL did not use the shared cache among neighboring nodes.

The medium competition also impacts the delivery delay. For all mechanisms, the delivery delay increased as the content request rate increases, as expected. LFBL produced the highest delay compared with our mechanisms and the default NDN. The reason for this higher delay is related to LFBL characteristic of using independent flows for the request and delivery of contents. This requirement of independent flow overloads nodes placed at the regions closest to producers. These nodes have to forward more interest and content packets than others. The consequence is the high delivery delay experienced by content packets. The overhead of interest packets on the network center creates overloading near to the destination nodes on LFBL. This occurs because nodes on the network center

present more neighbors than the peripheral nodes. This overload increased the packets' hop counts by forcing part of the flows to use peripheral nodes to deliver the content packets.

The proposed mechanism, PIF with $p = 0.2$ (PIF02 for the sake of brevity), presented the highest performance in terms of overhead. With PIF02, high-centrality nodes only forward 20% of interest packets received, as explained in Section 4. This mechanism produces a lower level of broadcast of interest packets, thus reducing the competition for the wireless channel. Therefore, PIF02 is more suitable for this scenario.

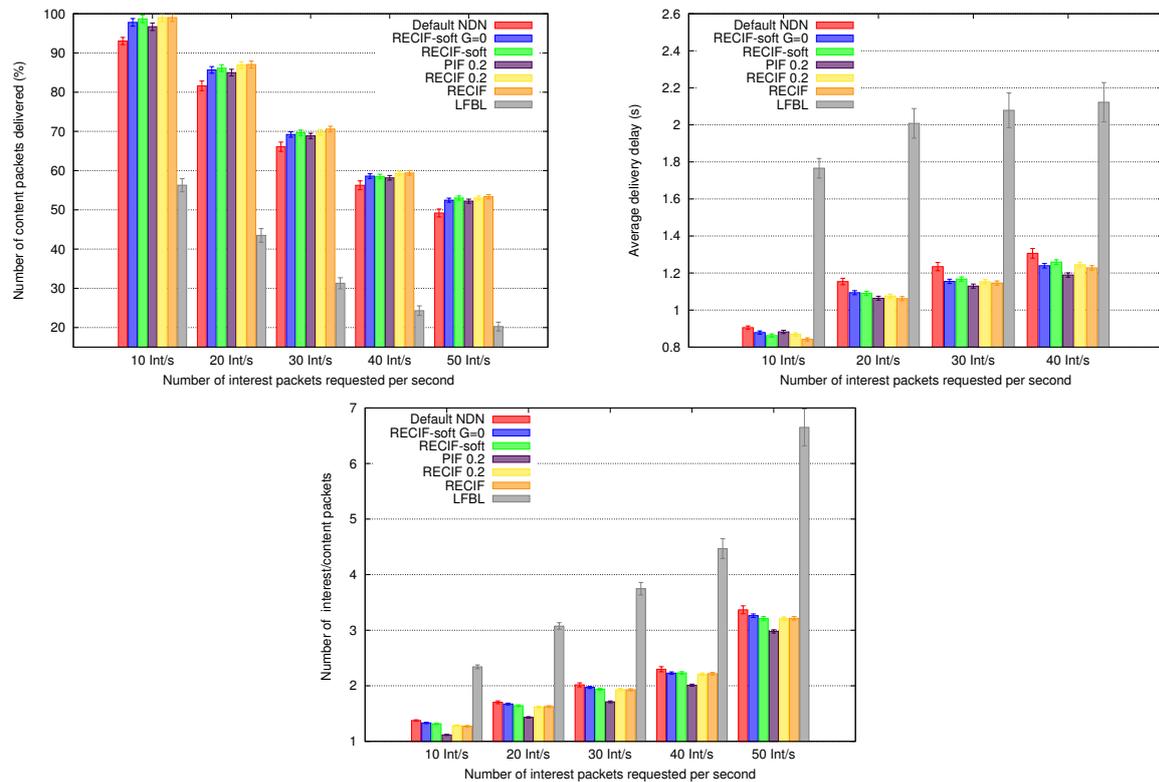


Figure 8. Content packets delivered, average delivery delay, and overhead for 30 consumers requesting popular contents at different rates.

5.3.3. The Impact of the Number of Consumers

After varying the content request rate, we conducted experiments to evaluate the impact of the number of consumers on the network. The reason for this evaluation was to analyze the performance of the proposed mechanisms with different network saturation levels caused only by the increasing number of consumers. We set the content request rate at 50 int/s for each consumer. The goal was to observe whether the network would experience a high saturation level, even for a low number of consumers.

Figure 9 presents the results. It can be observed that ReCIF and ReCIF02 provided higher delivery rates as the number of consumer nodes increased. This fact, discussed before, is related to the less restrictive criteria of these mechanisms which therefore, enables them to take advantage of in-network caching by retrieving the most popular contents from their near neighbors. In this scenario, the PIF02 restrictive strategy also produced a lower level of broadcast of interest packets. Therefore, it was not the best mechanism regarding content delivery but provided less overhead.

The performance of the LFBL degraded as more consumers were added to the network. This is a consequence of the multiple flows created by LFBL between consumers and producers. After the request phase, consumers retrieve contents directly from producers. Thus, LFBL sets multiple flows per consumer, creating multiple flows between consumers and producers. Hence, LFBL presented the worst performance in terms of delivery rate and average delay for 30 consumers.

For two consumers requesting contents by popularity, LFBL presented a similar performance to our mechanisms concerning the content delivery rate. The reason for this is because with two consumers, there would be only two flows from consumer to producers. However, as the number of consumers increases, our mechanisms provided higher levels of performance. This performance gain occurred because our mechanisms were able to use the shared cache efficiently.

In this scenario, PIF02 presented the best performance when compared to our other mechanisms. The reason for this is the same as that described for the previous scenario: PIF02 delivers contents at a slightly lower rate than ReCIF with less overhead and less delay.

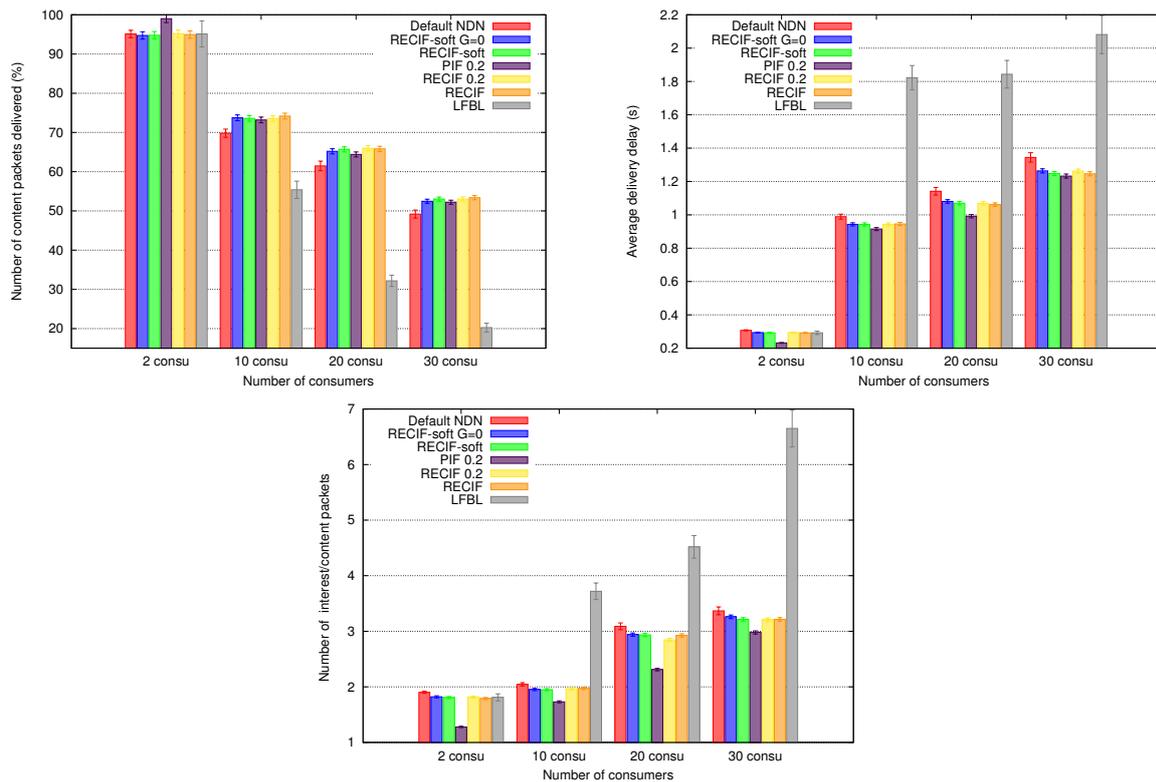


Figure 9. Content packets delivered, average delivery delay and overhead for 2, 10, 20 and 30 consumers requesting popular content at 50 interests/s.

5.3.4. The Impact of Cache Size

We argue, based on previous results, that NDN has the best performance against LFBL due to the more efficient use of in-network caching. To verify this hypothesis we conducted experiments considering a scenario with two consumers requesting contents, and we varied the cache size of consumers. The reason for considering two consumers lies in the fact that, for LFBL, the higher the number of consumers is, the higher the competition for different flows is. Therefore, with two consumers, we tried to minimize interflow interference. We varied the cache size of nodes from a content of 10,000 to 50. We initially used the same cache size as that of previous simulations, 10,000. To reduce that cache size, we observed the mean number of PIT entries during previous experiments and thus, set the cache size to 200. The idea was to choose a starting value for the cache size limit, and thus decrease this size by half. The other cache size values were contents of 100 and 50.

In Figure 10, it can be observed that the PIF02 mechanism delivered more contents than the other mechanisms until the cache size reached 100, because more contents were retrieved from the caches of the nearest nodes. If the cache size was less than 100, PIF02 started to deliver fewer content than all the other mechanisms. The reason for that resulted from the combination of the PIF02 forwarding criteria and fewer cache hits from near neighbors. Namely, fewer neighbor nodes had popular contents in

their small cache sizes. Therefore, this reduced the probability of receiving content from neighbors. Hence, this restriction promoted a smaller average delivery delay for PIF02 with a drawback of less content delivery. This behavior occurred because the farthest nodes used less shared cache from their neighbors.

Figure 10 also shows that LFBL outperformed the default NDN and our protocols for a cache size equal to 50 kB with few consumers. However, it is worth mentioning that this extreme cache limitation was only induced to prove our hypothesis. For example, in our configuration, each content was 1 KB and thus, a cache that stored 50 contents only required 50 KB of memory space. This cache is not a hard requirement for current network devices.

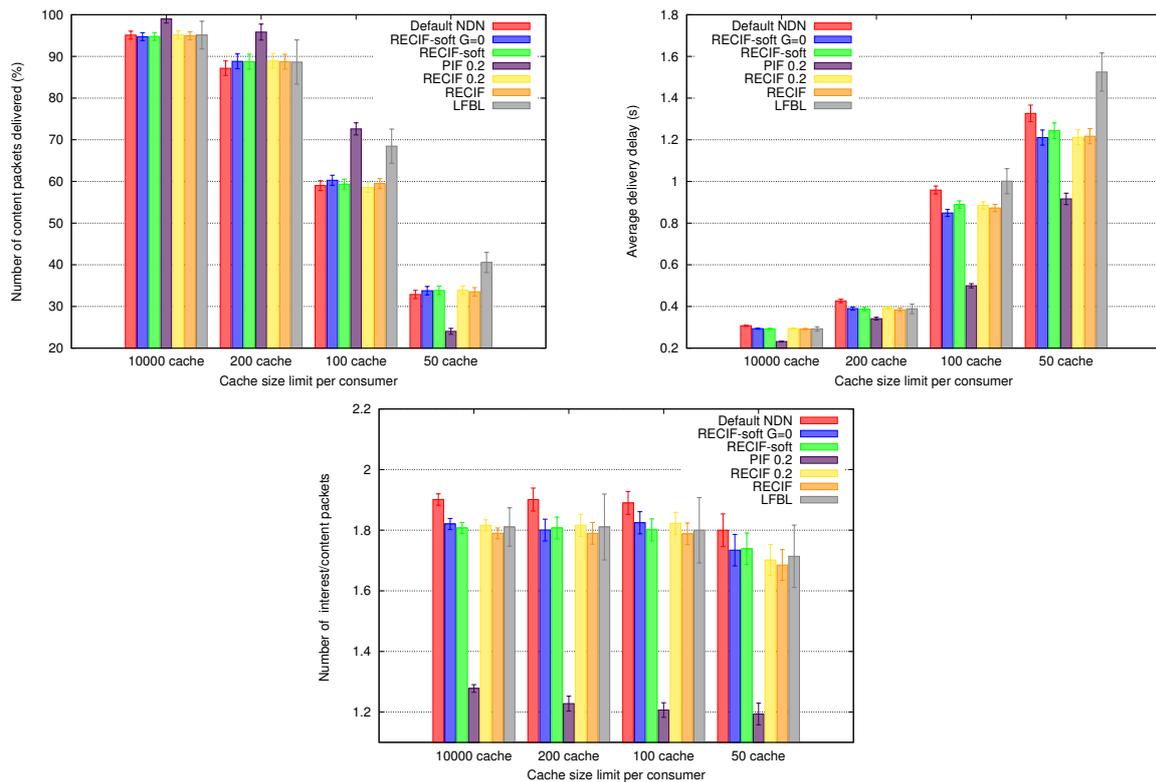


Figure 10. Content packets delivered, average delivery delay, and overhead for two consumers requesting popular content with different cache sizes.

5.3.5. The Impact of the Number of Nodes

We conducted tests on the Portland scenario with fewer network nodes. Accordingly, we removed nodes located on the right side of the river, as indicated in Figure 5. The idea was to evaluate the performance of our mechanisms in less dense scenarios. For this scenario, we considered eight consumers and two producer nodes and increased the number of interest requests from 10 to 50 int/s.

Figure 11 shows that RECIF-soft outperformed the other mechanisms in terms of the delivery rate. This mechanism reduced the number of interest packets forwarded, but it was less aggressive than PIF02, for example. RECIF-soft delivered more contents than the default NDN when the interest rate increased. For 50 int/s, RECIF-soft delivered 21.9% and 150.6% more contents than the default NDN and LFBL respectively. RECIF-soft also reduced the delivery delay compared with the default NDN and LFBL. Hence, RECIF-soft achieved a better adaptive retransmit threshold to restrict the number of interest packets broadcast on the network. Therefore, RECIF-soft delivered more contents by using an adaptive interest restriction. This restriction can be seen in Figure 11, where RECIF-soft presents less overhead than the default NDN but has a higher overhead than PIF02. More aggressive

mechanisms reduce the number of interest packets that reach the producer and therefore, content delivery is reduced. In this less saturated scenario, this restrictive approach is not ideal.

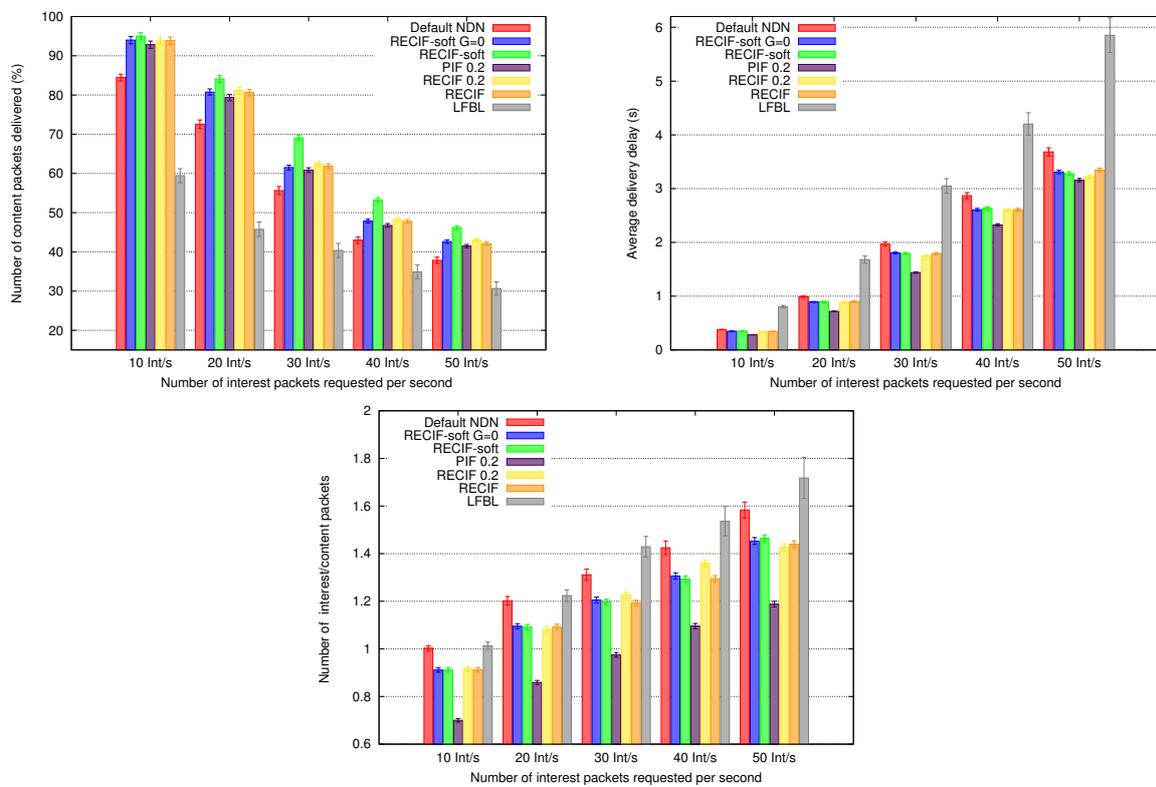


Figure 11. Content packets delivered, average delivery delay, and overhead for eight consumers requesting popular content at different rates.

The PIF, ReCIF, and ReCIF+PIF mechanisms provide (i) a higher content delivery as the network saturation level increases, (ii) less broadcast of interest packets to retrieve contents, and (iii) a lower average delivery delay. Nevertheless, some mechanisms are more suitable for less saturated environments, such as ReCIF, and others, like PIF02, performs better in a saturated environment. The main reason for these behaviors is the interest forwarding criteria used by the mechanisms. As a result, more restrictive criteria tend to be suitable for saturated networks. On the other hand, for less saturated scenarios, the RECIF-soft mechanism presents a higher performance because its restriction criteria adapt better to the scenario. This flexible filter criteria, present in RECIF-soft, allow higher content delivery without the decreased overhead performance present in PIF02. Thus, less restrictive and more flexible criteria are suitable for less saturated networks.

6. Conclusions

One of the main challenges in named-data wireless mesh networks is the broadcast storm problem, which is caused by the forwarding of interest packets. With NDN, each node broadcasts an interest packet received for the first time. Thereafter, every neighbor will repeat the same procedure, leading to a broadcast storm of interest packets. The main reason for this behavior lies in the fact that wireless nodes typically have only one network interface. Thus, interface-based control strategies that are used in wired NDNs to reduce the broadcast storm are not able to work properly on NDN-based WMNs.

We proposed, in this paper, a set of mechanisms that limit the forwarding of interest packets to reduce the negative effects of the broadcast storm in WMNs. The goal of the proposed mechanisms is to decrease the number of interest packets forwarded and thus, to decrease the probability of collisions caused by these packets. The first proposed mechanism is called Probabilistic Interest Forwarding (PIF). With PIF, high-centrality nodes forward packets with a probability of p . Therefore, PIF reduces interest

packet propagation on central nodes by reducing the number of packets that a central node may forward. With the second one, called Retransmission-Counter-based Interest Forwarding (ReCIF), nodes consider the number of hops traversed by an interest packet to forward this packet or not based on the retransmission threshold. ReCIF also has two operation modes, hard and soft, which are both used to define this threshold. The third proposed mechanism combines the criteria of PIF and ReCIF and is referred to as ReCIF+PIF. The idea is to guarantee that interest packets forwarded fewer times have priority to be forwarded by a node, and other interest packets are forwarded with a probability of p .

Our mechanisms were compared with both the default NDN forwarding mechanism and the Listen First Broadcast Later (LFBL) protocol through simulation. The results indicated that our mechanisms reduce the amount of interest packets forwarded by nodes to retrieve contents. As a consequence, PIF, ReCIF, and ReCIF+PIF provide higher rates of content delivery as the network saturation level increases and lower average delivery delay. We also observed that some mechanisms are more suitable for low-contention levels, such as ReCIF, and others, like PIF02, perform better for high-contention levels. The main reason for this behavior is the interest forwarding criteria used by the mechanisms.

We also conclude that our proposed mechanisms are more efficient in scenarios with a higher number of hops between source and destination. For these scenarios, our mechanisms outperformed the default NDN forwarding mechanism by up to 21% regarding the data delivery rate and provided a 25% lower delivery delay. Our mechanisms provided less delay when delivering contents to consumer nodes when compared to default NDN and LFBL. In less and highly saturated scenarios our mechanisms dynamically adapt to provide a higher performance using the shared cache and reducing the overhead.

For future works, we intend to test our mechanisms in other scenarios, including scenarios that present user mobility. The idea is to test our mechanisms in scenarios where pedestrian mobile users request and provide content using WMNs. We will also conduct tests on vehicular networks and compare our mechanisms with other specific protocols for this kind of network.

Author Contributions: Both authors have the same contribution to this paper.

Funding: The authors would like to thank CNPq, CAPES, FAPERJ, Proppi/UFF, TBE/ANEEL, CELESC/NeoDomino/ANEEL, and TAESA/ANEEL for the financial support to this research and development work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Canali, C.; Renda, M.E.; Santi, P.; Bursi, S. Enabling efficient peer-to-peer resource sharing in wireless mesh networks. *IEEE Trans. Mob. Comput.* **2010**, *9*, 333–347. [[CrossRef](#)]
2. Yang, Y.; Wang, J.; Kravets, R. Designing routing metrics for mesh networks. In Proceedings of the IEEE Workshop on Wireless Mesh Networks (WiMesh), Santa Clara, CA, USA, 26 September 2005.
3. Pathak, P.H.; Dutta, R. A survey of network design problems and joint design approaches in wireless mesh networks. *IEEE Commun. Surv. Tutor.* **2011**, *13*, 396–428. [[CrossRef](#)]
4. Al-Arnaout, Z.; Fu, Q.; Frean, M. A content replication scheme for wireless mesh networks. In Proceedings of the 22nd international workshop on Network and Operating System Support for Digital Audio and Video, Toronto, ON, Canada, 7–8 June 2012; pp. 39–44.
5. Oh, S.Y.; Lau, D.; Gerla, M. Content centric networking in tactical and emergency manets. In Proceedings of the Wireless Days (WD), Venice, Italy, 20–22 October 2010; pp. 1–5.
6. De Brito, G.M.; Velloso, P.B.; Moraes, I.M. *Information Centric Networks: A New Paradigm for the Internet*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
7. Torres, J.V.; Alvarenga, I.D.; Boutaba, R.; Duarte, O.C.M. An autonomous and efficient controller-based routing scheme for networking Named-Data mobility. *Comput. Commun.* **2017**, *103*, 94–103. [[CrossRef](#)]
8. Li, Z.; Simon, G. Cooperative caching in a content centric network for video stream delivery. *J. Netw. Syst. Manag.* **2015**, *23*, 445–473. [[CrossRef](#)]

9. Kim, D.; Kim, J.H.; Moon, C.; Choi, J.; Yeom, I. Efficient content delivery in mobile ad-hoc networks using CCN. *Ad Hoc Netw.* **2016**, *36*, 81–99. [[CrossRef](#)]
10. Meisel, M.; Pappas, V.; Zhang, L. Ad hoc networking via named data. In Proceedings of the Fifth ACM International Workshop on Mobility in the Evolving Internet Architecture, Chicago, IL, USA, 20–24 September 2010; pp. 3–8.
11. Jacobson, V.; Smetters, D.K.; Thornton, J.D.; Plass, M.F.; Briggs, N.H.; Braynard, R.L. Networking named content. In Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, Rome, Italy, 1–4 December 2009; pp. 1–12.
12. Bosunia, M.R.; Kim, A.; Jeong, D.P.; Park, C.; Jeong, S.H. Enhanced Multimedia Data Delivery based on Content-Centric Networking in Wireless Networks. *Appl. Math.* **2015**, *9*, 579–589.
13. Sandvine Top 10 Peak Period Applications. Available online: <https://www.sandvine.com/pr/2016/6/22/sandvine-report-netflix-encoding-optimizations-result-in-north-american-traffic-share-decline.html> (accessed on 15 July 2017).
14. Amadeo, M.; Campolo, C.; Molinaro, A. Content-centric networking: Is that a solution for upcoming vehicular networks? In Proceedings of the Ninth ACM International Workshop on Vehicular Inter-Networking, Systems, and Applications, Ambleside, UK, 25–29 June 2012; pp. 99–102.
15. Amadeo, M.; Molinaro, A.; Campolo, C.; Sifalakis, M.; Tschudin, C. Transport layer design for named data wireless networking. In Proceedings of the 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPs), Toronto, ON, Canada, 27 April–2 May 2014; pp. 464–469.
16. Akyildiz, I.F.; Wang, X.; Wang, W. Wireless mesh networks: A survey. *Comput. Netw.* **2005**, *47*, 445–487. [[CrossRef](#)]
17. Carofiglio, G.; Gallo, M.; Muscariello, L. ICP: Design and evaluation of an interest control protocol for content-centric networking. In Proceedings of the 2012 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPs), Orlando, FL, USA, 25–30 May 2012; pp. 304–309.
18. Liu, D.; Chen, B.; Yang, C.; Molisch, A.F. Caching at the wireless edge: Design aspects, challenges, and future directions. *IEEE Commun. Mag.* **2016**, *54*, 22–28. [[CrossRef](#)]
19. IEEE Computer Society LAN MAN Standards Committee. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. *IEEE Standard 802.11-1997* **1997**. [[CrossRef](#)]
20. Meisel, M.; Pappas, V.; Zhang, L. Listen first, broadcast later: Topology-agnostic forwarding under high dynamics. In Proceedings of the Annual Conference of International Technology Alliance in Network and Information Science, London, UK, 15–16 September 2010; p. 8.
21. Perkins, C.; Belding-Royer, E.; Das, S. Ad Hoc on-Demand Distance Vector (AODV) Routing. Technical Report; 2003. Available online: <https://tools.ietf.org/html/rfc3561> (accessed on 27 September 2018).
22. Chen, M.; Hao, Y.; Qiu, M.; Song, J.; Wu, D.; Humar, I. Mobility-aware caching and computation offloading in 5G ultra-dense cellular networks. *Sensors* **2016**, *16*, 974. [[CrossRef](#)] [[PubMed](#)]
23. Afanasyev, A.; Jiang, X.; Yu, Y.; Tan, J.; Xia, Y.; Mankin, A.; Zhang, L. NDNS: A DNS-Like Name Service for NDN. In Proceedings of the 26th International Conference on Computer Communications and Networks (ICCCN), Vancouver, BC, Canada, 31 July–3 August 2017.
24. Tatar, A.; de Amorim, M.D.; Fdida, S.; Antoniadis, P. A survey on predicting the popularity of web content. *J. Internet Serv. Appl.* **2014**, *5*, 1. [[CrossRef](#)]
25. Rainer, B.; Posch, D.; Hellwagner, H. Investigating the performance of pull-based dynamic adaptive streaming in NDN. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 2130–2140. [[CrossRef](#)]
26. Burrelli, S.; Canali, C.; Renda, M.E.; Santi, P. MeshChord: A location-aware, cross-layer specialization of Chord for wireless mesh networks (concise contribution). In Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom), Hong Kong, China, 17–21 March 2008; pp. 206–212.
27. Gibson, C.; Bermell-Garcia, P.; Chan, K.; Ko, B.; Afanasyev, A.; Zhang, L. Opportunities and Challenges for Named Data Networking to Increase the Agility of Military Coalitions. In Proceedings of the Workshop on Distributed Analytics Infrastructure and Algorithms for Multi-Organization Federations (DAIS), San Francisco, CA, USA, 4–8 August 2017.
28. Hannan, A.; Arshad, S.; Azam, M.A.; Loo, J.; Ahmed, S.H.; Majeed, M.F.; Shah, S.C. Disaster management system aided by named data network of things: Architecture, design, and analysis. *Sensors* **2018**, *18*, 2431. [[CrossRef](#)] [[PubMed](#)]

29. Chatzimisios, P.; Boucouvalas, A.; Vitsas, V. Effectiveness of RTS/CTS handshake in IEEE 802.11 a wireless LANs. *Electron. Lett.* **2004**, *40*, 915–916. [[CrossRef](#)]
30. Amadeo, M.; Molinaro, A. CHANET: A content-centric architecture for IEEE 802.11 MANETs. In Proceedings of the 2011 International Conference on the Network of the Future (NOF), Paris, France, 28–30 November 2011; pp. 122–127.
31. Schneider, K.M.; Mast, K.; Krieger, U.R. CCN forwarding strategies for multihomed mobile terminals. In Proceedings of the 2015 International Conference and Workshops on Networked Systems (NetSys), Cottbus, Germany, 9–12 March 2015; pp. 1–5.
32. Lu, Y.; Li, X.; Yu, Y.T.; Gerla, M. Information-centric delay-tolerant mobile ad-hoc networks. In Proceedings of the 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPs), Toronto, ON, Canada, 27 April–2 May 2014; pp. 428–433.
33. Rainer, B.; Posch, D.; Leibetseder, A.; Theuermann, S.; Hellwagner, H. A low-cost NDN testbed on banana pi routers. *IEEE Commun. Mag.* **2016**, *54*, 105–111. [[CrossRef](#)]
34. Zhao, N.; Liu, X.; Yu, F.R.; Li, M.; Leung, V.C. Communications, caching, and computing oriented small cell networks with interference alignment. *IEEE Commun. Mag.* **2016**, *54*, 29–35. [[CrossRef](#)]
35. Mascarenhas, D.M.; Moraes, I.M. Limiting the interest-packet forwarding in information-centric wireless mesh networks. In Proceedings of the Wireless Days (WD) 2014, Rio de Janeiro, Brazil, 12–14 November 2014; pp. 1–6.
36. Clausen, T.; Jacquet, P. *Optimized Link State Routing Protocol (OLSR)*. Technical Report; 2003. Available online: <https://tools.ietf.org/html/rfc3626> (accessed on 27 September 2018).
37. Akkaya, K.; Younis, M. A survey on routing protocols for wireless sensor networks. *Ad Hoc Netw.* **2005**, *3*, 325–349. [[CrossRef](#)]
38. Afanasyev, A.; Moiseenko, I.; Zhang, L. *ndnSIM: NDN Simulator for NS-3*; Named Data Networking (NDN) Project, Technical Report NDN-0005; 2012; Volume 2. Available online: <http://named-data.net/techreport/TR005-ndnsim.pdf> (accessed on 27 September 2018).
39. Senior, R.; Phillips, C. CRAWDAD Data Set Pdx/Metrofi (v. 2011-10-24). Available online: <http://crawdad.org/pdx/metrofi/> (accessed on 27 September 2018).



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).