Closing the Floodgate with Stateless Content-Centric Networking

Cesar Ghali[1+], Gene Tsudik[1], Ersin Uzun[2], Christopher A. Wood[1*]
[1]Department of Computer Science, University of California Irvine, Irvine, CA, USA
[2]Computer Science Laboratory, Palo Alto Research Center, Palo Alto, CA, USA
Emails: cghali@uci.edu, gts@ics.uci.edu, ersin.uzun@parc.com, woodc1@uci.edu

***Abstract*—Information-Centric Networking (ICN) is a recent paradigm that claims to mitigate some limitations of the current IP-based Internet architecture. The centerpiece of ICN is named and addressable content, rather than hosts or interfaces. Content-Centric Networking (CCN) is a prominent ICN instance that shares the fundamental architectural design with its equally popular academic sibling Named-Data Networking (NDN). CCN eschews source addresses and creates one-time virtual circuits for every content request (called an interest). As an interest is forwarded it creates state in intervening routers and the requested content is delivered back over the reverse path using that state.**

Although a stateful forwarding plane might be beneficial in terms of efficiency and resilience to certain types of attacks, this has not been decisively proven via realistic experiments. Since keeping per-interest state complicates router operations and makes the infrastructure susceptible to router state exhaustion attacks (e.g., there is currently no effective defense against Interest Flooding attacks), the value of the stateful forwarding plane in CCN should be re-examined.

In this paper, we explore supposed benefits and various problems of the stateful forwarding plane. We then argue that its benefits are uncertain at best, and it need not be a mandatory CCN feature. To this end, we propose a new stateless architecture for CCN that provides nearly all functionality of the stateful design without its headaches. We analyze performance and resource requirements of the proposed architecture via experiments.

Keywords*—*Content-Centric Networking, Named Data Networking, Denial of Service, Interest-Flooding Attack, Stateless CCN

I. Introduction

Information-Centric Networking (ICN) [1] is a networking paradigm that emerged as an alternative to the host-based communication approach of the current IP-based Internet architecture. Content-Centric Networking (CCN) [2], [3] is one industry-driven instance of this approach. (It is closely related to Named-Data Networking (NDN), which can be viewed as CCN's academic dual.) While IP traffic consists of packets sent between communicating end-points, CCN traffic is comprises explicit requests for, and responses to, named content objects. These requests, called *interests*, refer to the desired content by name. An interest is forwarded by routers (using the name) towards a content producer until satisfied by the latter or by an on-path router that has a cached copy. The corresponding response, called *content*, is forwarded along the reverse path to the consumer. To reduce end-to-end latency and congestion, CCN routers may opportunistically cache content to satisfy future interests.

In CCN, neither interest nor content messages carry source addresses. In order to correctly deliver content to consumers, routers maintain per-interest state: for each pending interest, a dedicated entry in a so-called Pending Interest Table (PIT). This state information maps interest names to interfaces on which they arrived. Routers use the interfaces in PIT entries to forward content upon receipt. After a content is forwarded downstream, the corresponding PIT entry is flushed.

Another purpose of the PIT is to support *interest collapsing* – a feature designed for handling multitudes of nearly simultaneous interests for the same content. Whenever a router receives an interest for which it has a matching PIT entry, the arrival interface of the new interest is added to the existing entry and the interest is not forwarded further. This prevents duplicate interests from being sent upstream, thus lowering overall congestion. However, as we argue later, interest collapsing rarely occurs in practice.

Furthermore, stateful forwarding enabled by PITs is supposed to provide flow balance via path symmetry between interest and content messages. Consequently, information from the PIT, e.g., interest to content Round-Trip Time (RTT), can be used to develop better congestion control and traffic shaping mechanisms [4]–[7]. However, using a PIT for flow balance and in-network congestion control is quite problematic in practice. In fact, flow balance is a misnomer in the current CCN design due to the (potentially huge) disparity in sizes between interest and content messages. Likewise, there is ample evidence that congestion control and transport protocols are best deployed at receivers [8]–[10], due to flow imbalance and dynamic routing in CCN. Another claimed advantage of stateful forwarding is that it can aid routers when responding to congestion since they can make autonomous and intelligent forwarding decisions for interests. However, this per-link congestion information need not be stored in the PIT.

From the perspective of infrastructure security, the PIT prevents *reflection attacks* since content is always forwarded according to PIT entries [11]. However, this per-packet state is costly to maintain. Various attempts to improve the efficiency of PIT-based forwarding have been studied in the context of CCN and NDN [12]–[14]. None of these address the fundamental design issue that the PIT size grows linearly with the number of distinct interests received by a router. This means that a PIT is a resource that can be easily abused. In fact, malicious exhaustion of PIT space in the form of Interest Flooding (IF) attacks [11] remains an important open problem. In such attacks, adversaries flood routers with nonsensical (i.e., unsatisfiable) interests in order to saturate the available PIT space. Once a router reaches its maximum PIT capacity

[+]Work done while at UCI.

[*]This author is supported by the NSF Graduate Research Fellowship DGE-1321846.

it either (1) drops new incoming interests, or (2) removes existing entries to free resources for new incoming interests. Both options can adversely impact legitimate traffic.

Given that many claimed benefits are dubious and considering associated infrastructure security problems, it becomes hard to justify the need for PITs in CCN. In this paper, we comprehensively assess (in Section II) the stateful forwarding plane of CCN with respect to each claimed benefit. We show that many benefits are: (1) either unrealistic or infeasible in practice, (2) can be achieved by means other than stateful forwarding, or (3) so marginal that their value simply does not justify their overhead. We then present, in Sections III and IV, a new stateless architecture for CCN based on Routable Backward Names (RBNs). This new design can co-exist with the current CCN architecture (with PITs) or replace it entirely. The proposed stateless architecture is different from that of Mirzazad-Barijough, et al. [15], where content is forwarded using MPLS-like labels and not per-packet state. As we show in Section II, [15] still assumes pull-based communication as the preferred mechanism for all applications and enforces path symmetry for interests and content objects. After discussing the design, we then present experimental results in Section V which indicate that the new design still retains the essence and performance characteristics of CCN while successfully avoiding pitfalls of per-interest packet state. We conclude with a discussion of related work and a summary in Sections VI and VII, respectively.

II. Assessing the PIT

The PIT is a fundamental and mandatory component of the CCN forwarding plane. It is a tabular data structure that maps interest names and other metadata to arrival downstream interfaces to which content responses should be forwarded. The shape and size of this table is directly dependent on the traffic that is processed by a forwarder. [16] studied PIT dynamics and showed that the number of entries can range from fewer than 100 for edge routers with a small number of per-namespace flows, to over 10^6 in the network core. [14] designed a PIT implementation that requires only 37MiB to 245MiB to forward traffic at 100Gbps, which can scale to fit the needs of realistic traffic, according to [16].

In this paper, we do not question or discuss the *implementaton* of the PIT. Instead, we question justifications for its existence. Below, we argue that – aside from being unnecessary to support CCN-like communication – the PIT's presence raises more (serious) problems than it solves.We support our argument by systematically analyzing the following alleged PIT benefits:

1) Reverse-path forwarding
2) Infrastructure security
3) Flow and congestion control
4) Interest collapsing

We then show that all these benefits are either false, unnecessary, or very meager at best.

A. Reverse-Path Forwarding

A key tenet of CCN is that content is never sent to a consumer who previously did not issue an interest (i.e., does not have a pending interest) for this content. According to [2], since interests contain no source addresses, PITs are needed:

> "...to forward Content Objects from producers to consumers along the interest reverse path by leaving per-hop state in each router..."

We disagree with this statement for two reasons. First, network path symmetry is not guaranteed and should not be assumed. [17] demonstrated that route symmetry between the same flow on the Internet is lower in the core than at the edges. Several tier-1 and tier-2 networks were studied and it was shown that, due to "hot-potato-routing," flow asymmetry exceeds 90% in the core. Thus, symmetric path routing in the core appears to directly contradict today's practices that promote and exploit path asymmetry for better traffic distribution. Attempting to enforce symmetric data traversal appears to be a challenge from an economic perspective.

Second, pull-based communication with symmetric paths is not well-suited for *all* applications. While appropriate for scalable content distribution applications[1], it is substantially different from modern TCP/IP applications and protocols which rely on interactive sessions and bidirectional streams between endpoints. For instance, the WebSocket [18] protocol uses full-duplex TCP streams for clients and servers that engage in real-time, bidirectional communication. It is used by many popular interactive applications, such as multimedia chat and multiplayer video games. Two-way communication is not limited to Web protocols. Voice applications such as Skype [19] and peer-to-peer systems such as BitTorrent [20] rely on two endpoints which both produce and consume data, as part of the application.

Given the relative infancy of CCN and abundance of real-world applications that currently do not fit CCN's mold, it is difficult to argue that the pull communication model can satisfy all application needs. For example, even today, some existing CCN applications abuse interest messages to carry information from consumers to producers [21]. Other applications rely on consumers and producers to send interests to each other. NDN-RTC, a recently developed NDN video teleconference application, is one such example that supports such bidirectional communication between peers [22]. (We use NDN and CCN interchangeably here since both are equivalent in this context.)

Another emerging application design pattern is data transport via set synchronization. The NDN ChronoSync protocol is a prime example of this pattern [23]. Each ChronoSync user acts as *both* a producer and consumer. Consumers (members) issue *long-standing* interests to a group (common namespace) about specific data to be synchronized; These interests are routed to all members. When target data is changed by someone, this member satisfies previous interest(s) with a fingerprint of the data in a content object. Each member is then responsible for requesting updated content to synchronize with the others. This protocol is built on the fundamental assumption that pull-based data transmission is the only communication pattern.

Based on the trends of current TCP/IP applications and proposed design strategies for CCN-based protocols and applications, it seems clear that bidirectional communication is here to stay. For it to work, router FIBs need to contain prefixes for all end-points – not just producers. Therefore, all communicating parties need to obtain and use a routable prefix, which effectively serves as an address. As a consequence,

[1]Which some believe to be already well-served by today's CDNs.

forwarding information stored in a PIT becomes redundant and unnecessary.

B. Infrastructure Security

Denial of Service (DoS) attacks are a major threat to any network infrastructure. DoS attacks in today's Internet include: bandwidth depletion, DNS cache poisoning, black-holing and prefix hijacking, as well as reflection attacks. [11] shows how CCN (in the context of NDN) prevents these types of attacks. Out of all attack types considered, the PIT is needed only to prevent reflection attacks [24]. Since content is forwarded based on PIT entries, such attacks are impossible in CCN. However, forwarding content via the PIT is not the only way to prevent reflection attacks. If packets have a source address, the ingress filtering technique in [25] – whereby ISPs filter packets based on source addresses – would work equally well.

Despite its resilience to reflection attacks, CCN is susceptible to another major attack type known as Interest Flooding (IF) [11]. In one IF attack, a malicious consumer (or a distributed botnet) issues nonsensical interests[2] so as to overwhelm targeted routers and saturate their PITs. This is due to fact that such interests will not be satisfied by their respective producers leaving router PIT resources occupied. According to [16], the PIT size can exceed 10^6 as upstream paths become congested. The problem worsens if a malicious consumer and producer cooperate to target a specific router. Although several attempts to detect, mitigate, and prevent them have been made [26]–[32][3] each of them is effective against only a very naïve or weak attacker. Thus, IF attacks remain a daunting open problem with no solution in sight barring network architecture changes.

C. Flow and Congestion Control

[33] presented the first thorough argument in support of a stateful forwarding plane in the context of NDN. Due to their near-identical features, the same applies to CCN. The PIT can be used to record RTTs for interest and content exchanges, which, in turn, is useful for making dynamic forwarding decisions. For instance, if the RTT for a given namespace on a particular link becomes too high, that link might be congested and alternatives should be explored. This type of in-network congestion and flow control has been studied further in [4], [10], [34]–[36]. For instance, [36] propose a joint hop-by-hop (i.e., in-network) and receiver-based control protocol that relies on PIT-based RTT measurements for flows. In-network flow control allows routers to control flow closer to congested links, as opposed to performing the same by receivers.

However, according to [7], flow differentiation is a difficult challenge. One approach to "interest shaping" is by controlling the flow of data on upstream and downstream links independently of flows. This does not require any information from the PIT. Instead, it relies on knowledge of average interest and content size, link bandwidth, and interest arrival rates (or demand). Similar to [10], it also relies on receiver-driven flow control via an Additive-Increase-Multiplicative-Decrease window. [37] is another example of a receiver-driven flow control protocol for CCN. In contrast, [34] proposed a rate-based congestion control protocol that exploits the multi-path and stateful nature of CCN. Given these results, it is not clear where congestion control logic is most appropriate. Nevertheless, recent trends in the ICN research community show that pushing stateful control protocols towards receivers, rather than to network nodes, is a viable and attractive approach.

D. Utility of Interest Collapsing

[38] is the first to accurately model interest collapsing in CCN and NDN. The results indicate that collapsing occurs very little, i.e., with probability rarely exceeding 0.15, at the edge of the network (where content will be cached) for popular content classes. The independent analysis we provide in [39] confirms these results. This means that, when caching is present at the edge, interest collapsing becomes almost useless in practice.

III. Stateless CCN using Backwards Routable Names

Based on the previous discussion, PITs are unnecessary to provide many of their offered services and simultaneously come at the price of serious infrastructure security problems that have not been addressed. To this end, we introduce a modified CCN architecture without PITs, called stateless CCN.

The main idea behind our stateless CCN design is simple: an interest now includes a new field called Backwards Routable Name (BRN), a routable prefix, similar to an IP source address. BRNs exist in a global namespace much like an IPv6 address. A BRN indicates *where* the corresponding content should be delivered. The corresponding content carries the BRN as its routable name towards the origin of the interest. Thus, with properly configured FIB entries, content is correctly delivered to the origin of the interest.[4] This modification to the CCN architecture is clearly inspired by IP – all packets (interest and content) are forwarded based on addresses they carry and not on network state. However, as we show below, this does not violate CCN's core value of named data being moved through, and stored in, the network.

To illustrate BRN-based forwarding, consider a scenario where a consumer Cr with *topological name* `/edu/uci/ics/gateway/bob` (N_{Cr}) requests content from a producer P with the name `/bbc/news/today` (N_{bbc}).[5] In this case, Cr is the origin of the interest. (As we will show later, it is not mandatory for a consumer to be an origin.) Let $Int[N, SN]$ be an interest with the routable name $N = N_{bbc}$ and Supporting Name $SN = N_{Cr}$. Also, let $C[N, SN]$ be the corresponding content object that matches $Int[N, SN]$ where $C.N = Int.N$, and $C.SN = Int.SN$. In this example, assume that $C[N, SN]$ is not cached anywhere.

1) Cr advertises its name N_{Cr} and the routing protocol propagates this information accordingly.
2) Cr issues $Int[N_{bbc}, N_{Cr}]$.
3) The network forwards $Int[N_{bbc}, N_{Cr}]$ towards P according to router FIB entries. At every hop, each router may optionally modify N_{Cr} if needed to preserve routing correctness and consumer privacy (see Section IV).

[2]For example, an interest with a name reflecting a valid producer's prefix, with a random number as its last segment.

[3]For details, see Section VI below.

[4]This requires origins to publicly advertise their BRN prefixes and participate in routing.

[5]Names are encoded using the Labeled Content Identifier (LCI) schema [40]. LCI names are the concatenation of individual name segments, separated by the `'/'` character, in a typical URI-like format.

```
Message := MessageType PacketName [Payload] [Validation]
MessageType := Interest | ContentObject | ...
PacketName := Name SupportingName
Name := CCNx Name
SupportingName := CCNx Name
Payload := OCTET+
Validation := ValidationAlg ValidationPayload
```

Fig. 1. Stateless Packet Format in ABNF; `ValidationAlg` and `ValidationPayload` elements are defined in [41].

4) Once P receives $Int[N_{bbc}, N_{Cr}]$ it replies with $C[N_{bbc}, N_{Cr}]$.
5) Similarly to Step 3, the network forwards $C[N_{bbc}, N_{Cr}]$ back to Cr, based on N_{Cr}, using the same interest forwarding strategy.

Several modifications need to be made to the existing CCN architecture and protocol to enable this communication. At a minimum, interest and content object messages should carry two names: one of the requested content and the other of the origin. Contrary to IP, these two names *do not* correspond to a source and destination address. The origin's name serves as a topological address to which the content object should be sent, whereas the data's name serves as a topology-agnostic locator and identifier for the data. Therefore, the addition of this name does not violate the core CCN value that data names are distinct and independent of network locale.

We suggest modifying both interest and content headers to include a new field called `SupportingName` (SN). This field contains the BRN of the interest origin. In the above example, interest and content headers would contain `/cnn/news/today` and `/edu/uci/ics/bob` as N and SN, respectively. Note that content object signatures can be generated in advance by omitting the content's SN field since this is only used for routing purposes. The resulting packet formats are shown in Figure 1 in ABNF form.

Currently, interest and content messages are very similar in CCN. Both contain a Name, Payload, and optional Validation fields [41]; they only differ in the top-level type. The stateless variant we propose still requires this distinction since interests and content objects are processed differently. For example, a router first attempts to satisfy an interest from its cache, while content is (optionally) cached prior forwarding.

We stress that a content might not follow the reverse path of the proceeding interest due to routing table configurations. In fact, we anticipate that origins might structure BRNs to control the degree of path asymmetry between interest and content messages.

Modified interest and content formats coupled with removing the PIT simplifies fast-path processing. Algorithms 1 and 2 show how a router would process interest and content messages. CS-Lookup represents a CS lookup operation based on N (content name). For clarity's sake, we omit content verification details in all algorithms. Interest forwarding involves a CS miss and FIB lookup whereas content object forwarding involves a CS update and FIB lookup. This is significantly simplified when compared to the traditional forwarding logic wherein interest forwarding requires a CS and PIT miss, PIT insertion, and FIB lookup whereas content object forwarding involves a CS miss, PIT hit and deletion, and CS update.

Algorithm 1 Process-Interest

1: **Input:** Interest $Int[N, SN]$, arrival interface F_i, CS, FIB
2: $C = \text{CS-Lookup}(\text{CS}, N)$
3: **if** $C \neq \text{nil}$ **then**
4: (Optionally) Modify SN to add privacy.
5: **Forward** C to F_i
6: **else**
7: prefix, $F_o = \text{FIB.Lookup}(N)$
8: **Forward** $Int[N, SN]$ to F_o based on local strategy
9: **end if**

Algorithm 2 Process-Content-Object

1: **Input:** Content Object $C[N, SN]$, CS, FIB
2: Cache $C[N, SN]$ with N as the key
3: $F_o = \text{FIB.Lookup}(SN)$
4: **Forward** $C[N, SN]$ to F_o based on local strategy

IV. Architecture Assessment

Despite significant research progress over the past five years, the PIT no longer seems to be a practical solution for content object forwarding in CCN. As discussed earlier, router PITs are prone to DoS (specifically IF) attacks. They also store information already available from FIBs (consumer routable prefixes) and enforce unnatural path symmetry in an increasingly asymmetric Internet. (The latter problems remain for the stateless CCN variant of Mirzazad-Barijough et al. [15].) The proposed stateless CCN variant mitigates these problems by specifying the use of source and destination prefixes. To support our claims, we compare the stateful and stateless CCN architectures with respect to aforementioned features. We then discuss both advantages and disadvantages of stateless CCN.

A. Revisiting the PIT Benefits

Reverse-Path Routing. The proposed stateless CCN scheme requires FIBs to be updated to accommodate BRN prefixes advertised by consumers. It might seem, at first, that this would lead to a tremendous increase in FIB size. However, recall that CCN interest (and now, content) forwarding is based on LPM. In stateless CCN, consumers announce their BRNs only to their first-hop routers (e.g., an access point), which, in turn, combines all its consumers' BRNs and announces an aggregate prefix to neighboring routers, similar to the Border Gateway Protocol (BGP) route-aggregation feature [42]. We will revisit this aggregation feature later.

Also, path asymmetry between interest and content messages in stateless CCN is more compliant with networking and routing practices of today's Internet. As argued in Section II-A, ISPs are likely to adopt an architecture that agrees with their present business model.

Forwarding Overhead. Stateful CCN dictates that, when processing an interest, a router should, in the worst case: (1) attempt to satisfy the interest from its CS, (2) create or modify a PIT entry for the interest, and (3) perform a FIB lookup. Meanwhile, stateless CCN eliminates (2), which reduces the number of operations needed to forward interests. To better understand this reduction, consider the operations needed to forward packets in stateful CCN. For interests, both the PIT and CS must be indexed (separately or together as in [43]) using full interest names. This costs a single lookup plus an additional write (to create a new, or update an existing, PIT

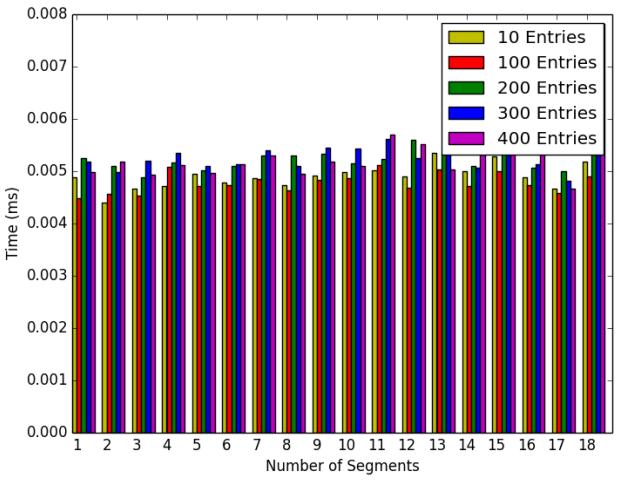


Fig. 2. Average PIT lookup and insertion overhead (in ms) as a function of the number of segments in a name.

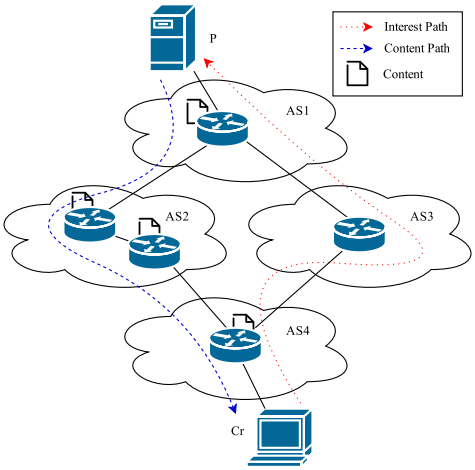


Fig. 3. Caching in stateless CCN. AS1 and AS4 are stub autonomous system representing tier-3 ISPs, AS2 and AS3 are transit autonomous system representing tier-1 ISPs.

entry) if the matching content is not cached. In stateless CCN, the PIT update procedure is removed, thereby improving the efficiency of the forwarding process.

To give an example of the overhead that is saved for this operation, we profiled the PIT lookup procedure for the PARC Metis forwarder [44]. Using a random set of URIs generated from the Cisco data set [45], we added and removed entries in the PIT at varying rates to match a desired steady state. We analyzed the PIT performance when its average number of entries is in the set $\{10, 100, 200, 300, 400\}$. The resulting lookup and insertion time is shown in Figure 2. For this implementation, running on a workstation with a 2.8 GHz Intel Core i7 CPU and 16GB of 1600 MHz DDR3 RAM running Ubuntu 14.04, we see that removing the PIT saves an average of approximately 4.5μs across all names in input data set.

Now consider content objects: forwarding requires a single PIT lookup, PIT deletion or eviction, and a CS write operation. In stateless CCN, the PIT index and update procedures are replaced with a FIB lookup procedure. Contrary to interest forwarding, stateless CCN content object forwarding should (in theory) be more expensive than that of stateful CCN. Using the data from [43], which presents a highly optimized software forwarder for NDN (with the same fundamental forwarding rules as stateful CCN), interests are forwarded at an average rate of 1500 cycles/packet whereas content objects are forwarded at an average rate of 550 cycles/packet. Interest processing requires: CS, PIT and FIB lookups as well as a PIT write operation to create or update an entry. Conversely, content processing requires PIT lookup and write (to remove an entry) operations.[6] The additional FIB lookup in interest processing is responsible for the extra overhead required for forwarding interests. Note that a FIB lookup is much slower than a PIT lookup. The reason is because the former is based on longest-prefix matching and actually consists of multiple lookups for different prefixes. This means that, in stateless CCN, replacing a PIT lookup while processing content with a FIB lookup should *increase* content forwarding overhead, but it would not exceed that of interest forwarding overhead in the standard stateful CCN design.

Flow and Congestion Control. Current receiver driven flow and congestion control algorithms are unaffected in stateless CCN. The only difference is that now routers are unable to compute the RTT for a given interest-content exchange. This prevents fine-grained flow control taking place close to congested links in the network. However, given that many flow control algorithms operate at the edge and do not rely on RTTs collected by routers, this is a tolerable loss.

B. Content Caching

As mentioned earlier, using BRNs for content routing does not preserve path symmetry. In fact, it encourages path asymmetry. Consequently, content might be cached along a different path than the interest originally traversed. It might seem that adjacent (or nearby) origins for the same content would therefore not benefit from in-network caching. We argue that this is not so. Due to their high processing rates, core routers will most likely not cache content. Meanwhile, consumer-facing routers would handle much less traffic and are thus more likely to cache content. In fact, caching has been shown to be most cost effective at the edges [46], e.g., at tier-3 ISP level. Since nearby consumers share the same edge router, they will all benefit from caching popular content in that router. This observation is supported by the results obtained in [17], wherein it is shown that path symmetry is highest at the edges of the network.

Figure 3 shows an example of caching in stateless CCN. The topology has 4 autonomous systems (AS-s). AS1 and AS4 are stubs representing tier-3 ISPs, while AS2 and AS3 are transits representing tier-1 ISPs.[7] Interests issued by Cr are forwarded towards P along the dotted (red) path, and content is forwarded back to Cr along the dashed (blue) path. Assuming that caching only occurs near the edges, content sent from P to Cr gets cached in AS4. Consequently, interests for the same content issued by other consumers in AS4 would be satisfied from AS4 cache(s).

[6]CS processing also includes a CS write operation to cache the received content. However, it can be done in parallel and not on the fast path.

[7]We ignore tier-2 ISPs for simplicity.

C. Infrastructure Security

We now discuss both beneficial and problematic infrastructure security issues in stateless CCN.

FIB Explosion. Stateless CCN necessitates that FIBs contain entries for origin and producer prefixes. Scalable name-based routing is still a topic of research for CCN and related architectures. Aggregation (as described later in this Section) helps reduce the number of entries in a FIB if those entries are topological; it does not offer much help for producer prefixes which are, in theory, agnostic to topological information [47]. Fortunately, since BRNs are necessarily topological, they can be aggregated similar to the way in which IPv4 addresses are aggregated behind a NAT.

Interest Flooding. Stateless CCN mitigates this attack by eliminating its root cause – the PIT. Without per-request state in routers, this attack vector is removed. By and large, this is the primary benefit of stateless CCN.

Reflection Attacks. Interest and content path symmetry in CCN prevents reflection attacks. However, in stateless CCN, BRNs serve as a *de facto* source address in interest, and destination in content, messages. Thus, reflection attacks reappear. Fortunately, the ingress filtering technique described in [25] can be used to mitigate them.

Cache and Content Poisoning. Content authentication in stateless CCN is identical to that in the stateful CCN architecture. It is done by producers signing content objects or using Self-Cerftifying Names (SCNs) [48]. Regardless of the method, all content *must* be verified by consumers. However, verification is not mandatory for routers, for several reasons; see [48] for more details. Lack of in-network content verification opens the door for content poisoning attacks [49]. Moreover, due to possible path asymmetry in BRN-based content forwarding, content poisoning countermeasures that work in the current CCN architecture do not apply anymore.

The PIT enables a router to apply the so-called Interest-Key Binding (IKB) rule [48], whereby consumers and producers collaborate to provide routers with enough (minimal) trust information to perform content verification. This information is currently stored in the PIT. However, as mentioned above, path asymmetry renders the IKB impractical *for the initial data request*. In stateless CCN, a router might receive (unsolicited) content without prior interest traversing the same path. If such content is returned on a path different from the original interest, routers cannot trust any information it carries. However, this does not prevent a router from opportunistically caching content it forwards. In doing so, the router can apply the IKB rule to *subsequent* requests for the *same cached data* without forwarding the interest upstream. (The difference here is that, in stateful CCN, the IKB rule can be applied to verify content before it is inserted into the cache, whereas now it must be applied once, and only once, the first cache hit occurs.)

Origin Privacy. Lack of source addresses in stateful CCN enables a degree of consumer privacy. If origins are consumers, then BRNs in stateless CCN negate this benefit much in the same way that global IPv6 addresses harm user privacy [50]. (However, as we will discuss, in an ideal deployment of stateless CCN, origins would *not* be consumers.) To aid mitigate this problem, a router R can assign a random identifier to each of its downstream consumers to be used as part of their BRN and could overwrite the BRN in all ingress interests based on this pseudonym (in line 4 of algorithm 1). For example, instead of including an BRN as `/edu/uci/ics/consmerA`, the gateway could set the BRN as `/edu/uci/ics/<rand>`, where "rand" is a random string that is rotated on a regular basis. The procedure to modify a BRN based on the arrival interface at a router is detailed in algorithm 3. One important benefit of this strategy is that "rand" can be rotated at random and independent of other routers so that consumers BRNs do not appear fixed upstream, thus mitigating interest linkability [51].

Algorithm 3 Mask-BRN

1: **Input:** SN, arrival interface F_i, FIB, r
2: $(\mathsf{prefix}, \mathsf{F_d})$:= FIB.$\mathsf{Lookup}(SN)$
3: **if** $F_d = F_i$ **then**
4: $\quad \mathsf{index} := |\mathsf{prefix}|$
5: $\quad SN_{\mathsf{index}} = H(SN_{\mathsf{index}}||r)$
6: **end if**
7: **Return** SN

D. Deployment Issues

The intent of our stateless CCN architecture is to provide an alternative to the current stateful CCN. This does not mean that one must replace the other. In fact, as we have designed it, they can co-exist. Consider the following scenarios:

1) Cr includes a BRN (SN) in an interest and upstream routers forward it as necessary. Stateful routers create PIT entries and stateless routers do not. In both cases, the interest is forwarded according to the FIB using content name N. Upon receipt of a content message, a stateful router uses its PIT to forward the content downstream, while a stateless router does that using the FIB and SN. In this case, stateful forwarders simply ignore the SN fields in both interests and content objects. This makes the proposed stateless CCN backwards compatible with the current CCN architecture.
2) Cr issues an interest as per current CCN rules. If a stateless router receives such an interest, it generates a NACK indicating that the interest cannot be forwarded further. To handle this NACK, some downstream node must provide a BRN for the interest and re-forward it as needed. This node can be the consumer or an AS gateway (i.e., a router that can forward packets to and from other ASs) acting as the origin.

Any node that satisfies an interest must honor its version (stateless or stateful) when producing a response. For example, if a producer (or a caching router) receives an interest with an BRN, it must reply according to stateless CCN by keeping both N and SN in the corresponding content.

We envision a hybrid approach where stateless CCN is deployed at the network core and stateful CCN at the edge. This aligns well with the CCN edge-caching strategy [46] and current path asymmetry in the Internet's core [17]. Edge routers in consumer-facing ASs will possess both caches and PITs to aid with content verification. When consumers issue interests, they first traverse through stateful CCN routers in a consumer-facing AS. When they leave this AS, the gateway, acting as the interest origin, supplies a BRN before forwarding upstream. Such interests will not induce any PIT state at the network core.

This hybrid approach has several powerful advantages. **First**, consider the benefits of the hybrid deployment with

respect to congestion control and mobility. If stateful CCN is deployed near the edge, then fine-grained congestion information can be collected and conveyed to consumers to adjust their transport protocol state accordingly. Moreover, as PITS are deployed in stateful CCN near the edge, where mobility events take place, existing proposals to handle mobility such as the trace-in-PIT proposal of [52] can be used. **Second**, it provides a native IF attack recovery mechanism. If R implements a PIT but does not have enough resources to create a new entry for Int, R can respond with a NACK similar to what is described above. Consumers, then, issue interests according to stateless CCN guidelines. The disadvantage of this approach as an effective IF attack countermeasure is that (1) it is reactive, so it can only be used after the attack occurs, and (2) it incurs an additional end-to-end latency since consumers (or downstream routers) need to reissue stateless CCN interests. **Third**, it allows forwarder state to scale where it scales best: *at the edge*. IF attacks are a problem specifically because the state does not scale well throughout the entire network. However, in smaller subnets, this state can be much better managed without falling victim to a DoS attack.

We also note that interests can cross stateless and stateful network boundaries with ease. If an interest Int travels from a stateful to a stateless network, the gateway must supply a BRN before forwarding the interest. The gateway is then considered the origin of the interest. Similarly, if an stateless interest arrives at a stateful gateway, the latter must store the BRN (in the SN field) in the corresponding PIT entry and *subsequently remove it from the interest*. This is necessary if Int is will cross, multiple times, across a stateful and stateless boundary.

V. Experiments and Analysis

We now evaluate performance of the stateless CCN in relation to stateful CCN. The key metric we use is the degree to which forwarding overhead is affected by stateless routing. To do this, we modified the ndnSIM 2.1 simulator [53], a simplified NDN implementation as an NS-3 [54] module, to support the stateless CCN architecture proposed in Section III. Specifically, we modified the NDN Forwarding Daemon (NFD) [55] to forward interests and content objects based on names and BRNs.

We then simulated topologies based on Deutsches ForschungsNetz (DFN), the German Research Network [56], [57] and AT&T core network (selected due to the size and diverse node distribution). Each topology consists of 160 consumers[8], a single producer connected to one of the edge routers, and multiple routers (more than 30). Each consumer generates 10 interests per second, with a random suffix so as to avoid cache hits. This is done to force interests to traverse the complete path to the producer and therefore maximize the amount of processing that takes place in forwarders in the upstream and downstream paths. This captures the worst-case scenario. In our experiments, neither the consumers nor the producer are equipped with a cache. We do, however, assess the forwarding overhead differences in the presence and absence of router caches.

The results of both experiments are shown in Figure 4. Figures 4(b) and 4(d) capture the overhead imposed by packet forwarding regardless of caching effects. They show that caching adds an overhead of approximately 20% to content processing; compare the blue lines in Figures 4(a) and 4(b), and Figures 4(c) and 4(d). Moreover, in both topologies, we observe that stateless packet forwarding imposes less overhead on routers compared to stateful interest and content forwarding. This is due to the fact that stateless packet forwarding does not require any PIT operations. The savings are quite significant, especially, for cache-less core routers that might process packets at rates of 100Gbps and over.

Furthermore, the overall content retrieval latency improves with stateless forwarding. Figure 5 shows a comparison of the RTT performance for both forwarders in the DFN topology. In this experiment, consumers always request unique content in order to avoid cache hits.[9]. On average, the content retrieval latency improves by more than 50%. The improvement reaches 77% for paths consisting of 6 hops. Although these results are dependent on the forwarder implementation inside ndnSIM, the results align with intuition and our previous experiment which show that stateless packet forwarding will, on average, improve due to the absence of the PIT.[10]

To justify this claim, we revisit the argument of Section IV-A which states that the forwarding cost for stateless content objects will be closer to that of stateful interests due to the former's need for a FIB lookup instead of a PIT lookup. To estimate this overhead, consider the forwarder in [43]. With a 64MB FIB and 2MB PIT, which can forward interests at a rate of approximately 1.2 MP/s (million packets per second), 2.3 MP/s, 4.2 MP/s, and 5.9 MP/s with 1, 2, 4, and 8 threads on a 2GHz core. In comparison, the content object forwarding throughputs are approximately 1.4 MP/s, 6.1 MP/s, 12.1 MP/s, and 14.2 MP/s under the same conditions. If the cost to forward content objects in the stateless variant is equal to that of interests, then the forwarding rate degrades by 14.3%, 62.3%, 65.3%, and 117.9%, respectively. These values capture the cost of the FIB lookup operation. However, given that stateless routers are no longer susceptible to DoS attacks, we deem this cost justified.

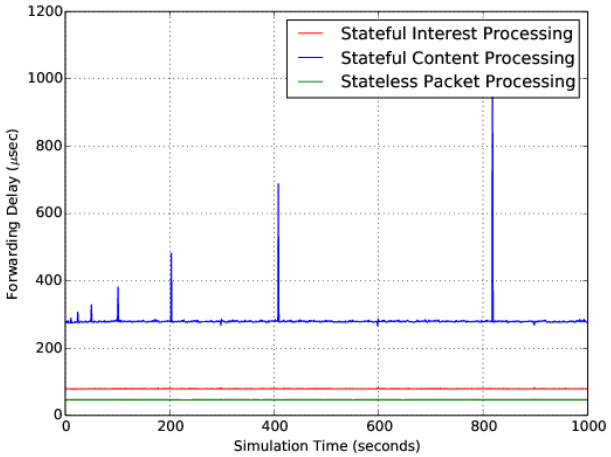
VI. Related Work

PIT-focused DoS attacks in CCN are a well-known problem [26]. Rate-based [27]–[29] and statistical-based tests [30]–[32] have been proposed to detect these attacks and subsequently limit the incoming interfaces upon which malicious interests arrive. However, this only treats a symptom of the problem– it does not solve the core issue of PIT state in routers. Dai et al. [58] propose a technique called "interest tracebacks" to identify malicious attackers and limit the rate at which they can send messages to the network. The key observation is that PIT state leaves a trace that terminates at the source of an interest. The network can use this trail to then identify the attacker. However, this approach depends on localized attackers sending interests at a high rate; it does not work for highly distributed adversaries. Similar in-network throttling techniques were discussed in [27] and [29]. Complementary to this general technique, Al-Sheikh et al. [59] introduce FIB

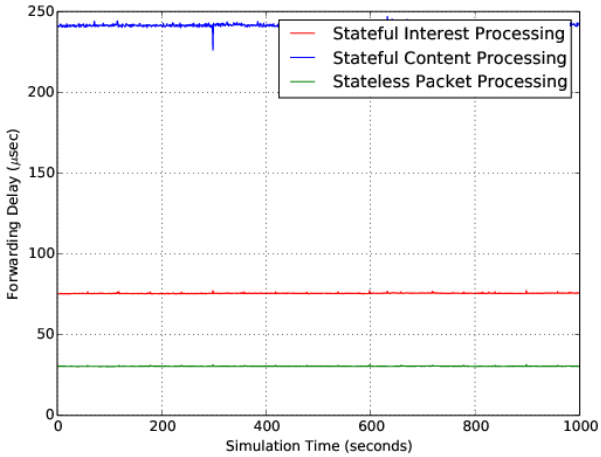
[8]Each consumer node in the figures consists of 10 actual consumers.

[9]We do not take caching into consideration to eliminate any randomize effects (caused by different eviction policies) on content retrieval latency.

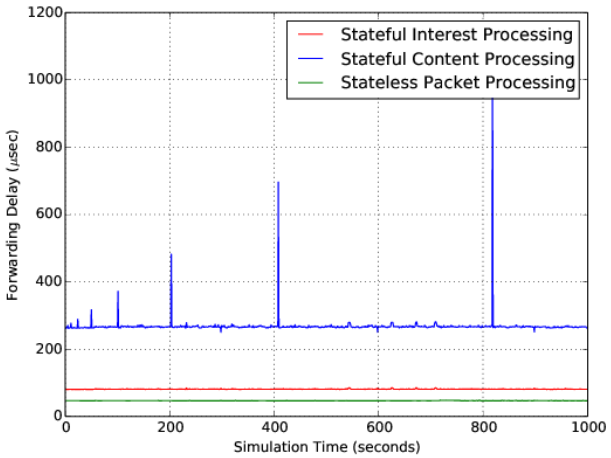
[10]We say on average since interests are forwarder more quickly whereas content objects necessarily require more time due to the FIB lookup in lieu of a PIT lookup.



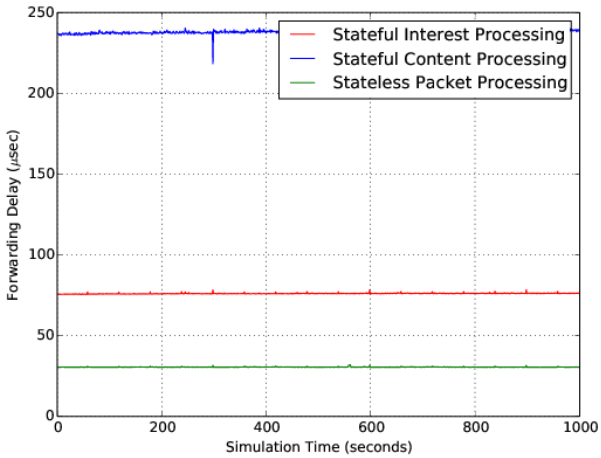
(a) Processing overhead for DFN topology with 160 consumers and routers with caches.



(b) Forwarding overhead for DFN topology with 160 consumers and routers without caches.



(c) Processing overhead for AT&T topology with 160 consumers and routers with caches.



(d) Forwarding overhead for AT&T topology with 160 consumers and routers without caches.

Fig. 4. Forwarding overhead in stateful (red, blue) and stateless (green) CCN variants.

exclude filters that seek to prevent malicious interests from propagating upstream to locations in the network where the requested content cannot possibly be served. These filters work for static content, only, and cannot be used to prevent interests for dynamic content from being forwarded. Li et al. [60] propose the use of consumer-based puzzles that must be solved as a native rate-limiting technique. These puzzles, or "interest cash," are generated by producers to be solved and must be completed *for each interest*. Although this approach is effective, it severely harms benign consumers.

Techniques to outright replace the PIT have also been proposed. [13] devised a "semi-stateful" solution wherein packets are marked (with Bloom Filters [61]) to be forwarded correctly. This approach shifts the state that was once in the PIT to the packets themselves and creates unnecessary communication and control overhead in the network. In a similar vein, Wang et al. [62] describe a protocol variant wherein resource-constrained PITs can offload the per-request state *into* interests that are forwarded. This technique puts PIT state "on the wire" and allows a PIT to naturally decrease in size as content is returned without dropping interests from benign consumers and routers. This is in contrast to our work where we defer state information to the routing protocol. Mirzazad-Barijough et al. [15] proposed a MPLS-like stateless variant for CCN. The authors approach solves the IF problem that stems from per-packet state but it still enforces path symmetry and does not generally aid applications that require bidirectional communication.

Salah et al. [63] used a router coordination framework called CoMon (Coordination with Lightweight Monitoring) to enable adjacent nodes to share information about forwarding state and traffic. Select routers are assigned the role of "monitor." The goal is to monitor interest and content exchanges and measure the (un)satisfaction rate. This information is periodically reported to a central "domain controller" that is in charge of processing the traffic reports to detect and respond

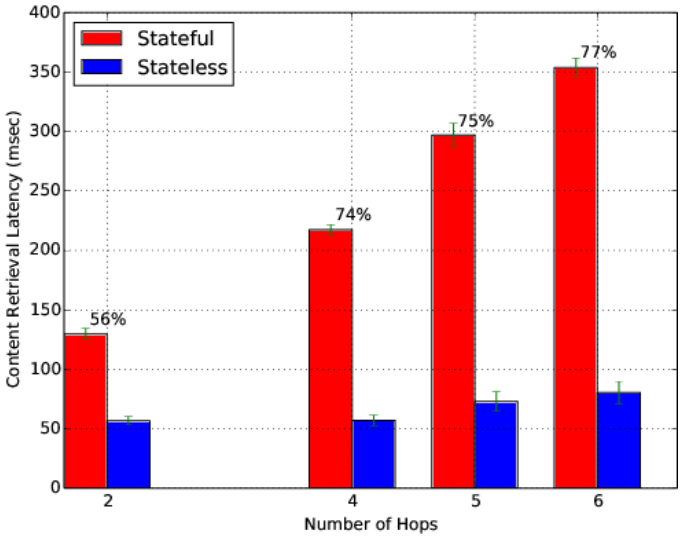


Fig. 5. Content retrieval latency as a function of number of hops between consumers and producers for both stateful and stateless forwarders. Note that paths with 3 hops do not exist in this topology.

to IF attacks. Monitoring routers are chosen based on their location in the network and closeness to producers. This solution assumes an unrealistic static topology and centralized post facto detection mechanism. In summary, this scheme is an extension to previous rate-based throttling approaches.

Almirshari et al. [64] proposed a technique to "piggy-back" interest and content objects to enable high throughput bidirectional communication in NDN. The authors approach introduces a new packet type in addition to interests and content objects. It also requires that interests are unnaturally extended to carry application data in the name. Moreover, this approach is still susceptible to IF attacks since it requires PIT state for bidirectional communication. Dai et al. [65] study extensions of PIT to support modern applications such as streaming services and online gaming. The proposed technique creates long-lived PIT entries to enable bidirectional communication between clients and servers. This only serves to make adversary's job easier in launching IF attacks.

Dabirmoghaddam. et al. [38] proposed an alternate probabilistic model for interest collapsing in CCN. The provided analytical and simulation results match and support what we presented in Section II-D. In [66] and [15], Garcia et al. introduce CCN-GRAM, which is a type of semi-stateless CCN architecture that uses "anonymous datagrams" to forward packets. These datagrams are essentially forwarded using a type of label swapping. Instead of storing state that is linear with respect to the number of interests, routers are required to store identifiers that correspond to input and output interface pairs. These identifiers are swapped in place as an interest is moved forward. CCN-GRAM achieves the same functionality as stateful CCN without per-packet state in routers. This means that it inherits the problems with forced path symmetry, which can be problematic for mobility (a growing use case). In contrast, our hybrid design permits PIT-based mobility solutions while still avoiding per-packet state where it is most costly – in the core.

VII. Conclusion and Future Work

Motivated by Interest Flooding attacks in current CCN, we proposed an alternative CCN architecture without PITs, called stateless CCN. We investigated the benefits of PIT and realized that they do not significantly improve the performance of content distribution. The proposed architecture is based on Routable Backward Names (RBNs) used to route content back towards requesting consumers. We provided a comprehensive performance and security assessment of the proposed stateless CCN architecture. We also discussed how it is practical to deploy and showed that deploying it alongside with current CCN does not achieve the expected benefits and performance.

However, removing the PIT came at the expense of losing support of some CCN features and extensions developed throughout the last few years. Consumer anonymity, is more difficult to achieve in RBN-based stateless CCN at the network layer without router participation or through the use of auxiliary protocols, such as ANDāNA [67] and AC3N [68]. Moreover, the Interest-Key Binding rule (IKB) [48] that enables content trust enforcement at the network layer, relies heavily on the PIT. Clearly, IKB cannot be applied in RBN-based stateless CCN. Nonetheless, we believe that advantages of the proposed architecture outweigh its drawbacks. We therefore defer solutions to the aforementioned disadvantages, e.g., trust, to future work.

VIII. Acknowledgment

The authors are grateful to ICCN'17 anonymous referees for theis useful comments. Christopher A. Wood was supported by the NSF Graduate Research Fellowship DGE-1321846.

References

[1] B. Ahlgren *et al.*, "A survey of information-centric networking," *IEEE Communications*, vol. 50, no. 7, 2012.

[2] V. Jacobson *et al.*, "Networking named content," in *CoNEXT*, 2009.

[3] I. Solis, "CCN 1.0 (tutorial)," in *ICN*, 2014.

[4] C. Yi *et al.*, "Adaptive forwarding in named data networking," *ACM CCR*, vol. 42, no. 3, 2012.

[5] J. Zhou *et al.*, "A proactive transport mechanism with explicit congestion notification for NDN," in *ICC*, 2015.

[6] H. Park *et al.*, "Popularity-based congestion control in named data networking," in *ICUFN*, 2014.

[7] Y. Wang *et al.*, "An improved hop-by-hop interest shaper for congestion control in named data networking," *ACM CCR*, vol. 43, no. 4, 2013.

[8] G. Carofiglio *et al.*, "Multipath congestion control in content-centric networks," in *INFOCOM WKSHPS*, 2013.

[9] S. Braun *et al.*, "An empirical study of receiver-based aimd flow-control strategies for CCN," in *ICCCN*, 2013.

[10] L. Saino *et al.*, "CCTCP: A scalable receiver-driven congestion control protocol for content centric networking," in *ICC*, 2013.

[11] P. Gasti *et al.*, "DoS and DDoS in named data networking," in *ICCCN*, 2013.

[12] W. So *et al.*, "Toward fast NDN software forwarding lookup engine based on hash tables," in *ANCS*, 2012.

[13] C. Tsilopoulos *et al.*, "Reducing forwarding state in content-centric networks with semi-stateless forwarding," in *INFOCOM*, 2014.

[14] H. Yuan *et al.*, "Scalable pending interest table design: From principles to practice," in *INFOCOM*, 2014.

[15] J. Garcia-Luna-Aceves and M. Mirzazad-Barijough, "Content-centric networking using anonymous datagrams," *arXiv preprint arXiv:1603.08491*, 2016.

[16] G. Carofiglio *et al.*, "Pending interest table sizing in named data networking," in *ICN*, 2015.

[17] W. John *et al.*, "Estimating routing symmetry on single links by passive flow measurements," in *IWCMC*, 2010.

[18] I. Fette *et al.*, "RFC 6455: The websocket protocol," 2011.

[19] "Skype," http://www.skype.com/.

[20] B. Cohen, "The BitTorrent protocol specification," 2008.

[21] J. Burke *et al.*, "Securing instrumented environments over content-centric networking: the case of lighting control and NDN," in *INFOCOM WKSHPS*, 2013.

[22] P. Gusev *et al.*, "NDN-RTC: Real-time videoconferencing over named data networking," in *ICN*, 2015.

[23] Z. Zhu *et al.*, "Let's chronosync: Decentralized dataset state synchronization in named data networking," in *ICNP*, 2013.

[24] P. Syverson, "A taxonomy of replay attacks [cryptographic protocols]," in *CSFW*, 1994.

[25] P. Ferguson, "RFC 2827: Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing," 2000.

[26] M. Virgilio *et al.*, "PIT overload analysis in content centric networks," in *SIGCOMM ICN Workshop*, 2013.

[27] A. Compagno *et al.*, "Poseidon: Mitigating interest flooding DDoS attacks in named data networking," in *LCN*, 2013.

[28] R. You *et al.*, "Detecting and mitigating interest flooding attack in content centric networking," *Advances in Computer Science and Technology*, vol. 65, 2014.

[29] A. Afanasyev *et al.*, "Interest flooding attack and countermeasures in named data networking," in *IFIP Networking*, 2013.

[30] N. T. Nguyen *et al.*, "Detection of interest flooding attacks in named data networking using hypothesis testing," 2015.

[31] T. Nguyen *et al.*, "An optimal statistical test for robust detection against interest flooding attacks in CCN," in *IFIP/IEEE IM*, 2015.

[32] J. Tang *et al.*, "Identifying interest flooding in named data networking," in *GreenCom*, 2013.

[33] C. Yi *et al.*, "A case for stateful forwarding plane," *Computer Communications*, vol. 36, no. 7, 2013.

[34] M. Mahdian, S. Arianfar, J. Gibson, and D. Oran, "Mircc: Multipath-aware icn rate-based congestion control," in *Proceedings of the 2016 conference on 3rd ACM Conference on Information-Centric Networking*. ACM, 2016, pp. 1–10.

[35] N. Rozhnova *et al.*, "An extended hop-by-hop interest shaping mechanism for content-centric networking," in *GLOBECOM*, 2014.

[36] G. Carofiglio *et al.*, "Joint hop-by-hop and receiver-driven interest control protocol for content-centric networks," in *SIGCOMM ICN Workshop*, 2012.

[37] Y. Ren *et al.*, "An interest control protocol for named data networking based on explicit feedback," in *ANCS*, 2015.

[38] A. Dabirmoghaddam *et al.*, "Characterizing interest aggregation in content-centric networks," *arXiv preprint arXiv:1603.07995*, 2016.

[39] C. Ghali, G. Tsudik, E. Uzun, and C. A. Wood, "Living in a pit-less world: A case against stateful forwarding in content-centric networking," *arXiv preprint arXiv:1512.07755*, 2015.

[40] M. Mosko, "CCNx 1.0 protocol specification roadmap," 2013.

[41] M. Mosko *et al.*, "CCNx messages in tlv format," 2015, https://tools.ietf.org/html/draft-irtf-icnrg-ccnxmessages-00.

[42] J. W. Stewart III, *BGP4: inter-domain routing in the Internet*. Addison-Wesley Longman, 1998.

[43] W. So *et al.*, "Named data networking on a router: fast and dos-resistant forwarding with hash tables," in *ANCS*, 2013.

[44] "Metis," https://github.com/parc/Metis, accessed: May 14, 2016.

[45] "The content name collection," http://www.icn-names.net/, accessed: April 8, 2016.

[46] J. Garcia-Luna-Aceves *et al.*, "Understanding optimal caching and opportunistic caching at the edge of information-centric networks," in *ICN*, 2014.

[47] T. C. Schmidt, S. Wölke, N. Berg, and M. Wählisch, "Let's collect names: How panini limits fib tables in name based routing," in *IFIP Networking Conference (IFIP Networking) and Workshops, 2016*. IEEE, 2016, pp. 458–466.

[48] C. Ghali *et al.*, "Network-layer trust in named-data networking," *SIGCOMM CCR*, vol. 44, no. 5, 2014.

[49] C. Ghali *et al.*, "Needle in a haystack: Mitigating content poisoning in named-data networking," in *NDSS SENT Workshop*, 2014.

[50] T. Narten, R. Draves, and S. Krishnan, "RFC 4941: Privacy extensions for stateless address autoconfiguration in IPv6," 2007.

[51] C. Ghali *et al.*, "Practical accounting in content-centric networking," in *NOMS*, 2016.

[52] Y. Zhang, H. Zhang, and L. Zhang, "Kite: A mobility support scheme for ndn," in *Proceedings of the 1st international conference on Information-centric networking*. ACM, 2014, pp. 179–180.

[53] S. Mastorakis *et al.*, "ndnSIM 2.0: A new version of the NDN simulator for NS-3," Technical Report, 2015.

[54] "Network simulator 3 (NS-3)," http://www.nsnam.org/.

[55] A. Afanasyev *et al.*, "NFD developers guide," Technical Report NDN-0021, NDN Project, Tech. Rep., 2014.

[56] "DFN-Verein," http://www.dfn.de/.

[57] "DFN-Verein: DFN-NOC," http://www.dfn.de/dienstleistungen/dfninternet/noc/.

[58] H. Dai *et al.*, "Mitigate ddos attacks in NDN by interest traceback," in *INFOCOM WKSHPS*, 2013.

[59] S. Al-Sheikh *et al.*, "Revisiting countermeasures against NDN interest flooding," in *ICN*, 2015.

[60] Z. Li *et al.*, "Interest cash: an application-based countermeasure against interest flooding for dynamic content in named data networking," in *CFI*, 2014.

[61] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *ACM Communications*, vol. 13, no. 7, pp. 422–426, 1970.

[62] K. Wang *et al.*, "Decoupling malicious interests from pending interest table to mitigate interest flooding attacks," in *GC Wkshps*, 2013.

[63] H. Salah *et al.*, "Lightweight coordinated defence against interest flooding attacks in NDN," in *INFOCOM WKSHPS*, 2015.

[64] M. Almishari *et al.*, "Optimizing bi-directional low-latency communication in named data networking," *SIGCOMM CCR*, vol. 44, no. 1, 2013.

[65] H. Dai *et al.*, "On pending interest table in named data networking," in *ANCS*, 2012.

[66] J. Garcia-Luna-Aceves, "New directions in content centric networking," in *Mobile Ad Hoc and Sensor Systems (MASS), 2015 IEEE 12th International Conference on*. IEEE, 2015, pp. 494–499.

[67] S. DiBenedetto *et al.*, "ANDaNA: Anonymous named data networking application," in *NDSS*, 2012.

[68] G. Tsudik, E. Uzun, and C. A. Wood, "Ac3n: Anonymous communication in content-centric networking," in *Consumer Communications & Networking Conference (CCNC), 2016 13th IEEE Annual*. IEEE, 2016, pp. 988–991.