Backscatter from the Data Plane – Threats to Stability and Security in Information-Centric Network Infrastructure

Matthias Wählisch

Freie Universität Berlin
Institut für Informatik
Email: waehlisch@ieee.org

Thomas C. Schmidt

HAW Hamburg
Dept. Informatik
Email: t.schmidt@ieee.org

Markus Vahlenkamp

HAW Hamburg
Dept. Informatik
Email: markus@vahlenkamp.net

Abstract

Information-centric networking (ICN) raises data objects to first class routable entities in the network and changes the Internet paradigm from host-centric connectivity to data-oriented delivery. However, current approaches to content routing heavily rely on data-driven protocol events and thereby introduce a strong coupling of the control to the data plane in the underlying routing infrastructure. In this paper, threats to the stability and security of the content distribution system are analyzed in theory, simulations, and practical experiments. We derive relations between state resources and the performance of routers, and demonstrate how this coupling can be misused in practice. We further show how state-based forwarding tends to degrade by decorrelating resources. We identify intrinsic attack vectors present in current content-centric routing, as well as possibilities and limitations to mitigate them. Our overall findings suggest that major architectural refinements are required prior to global ICN deployment in the real world.

Keywords: Security, vulnerability, performance evaluation, interest flooding, content-centric routing, denial of service (DoS)

Published in Computer Networks, Vol. 57, No. 16, pp. 3192–3206. *November 7, 2013*

1. Introduction

One major dedication of today's Internet is the global distribution of content in huge amounts. Content distribution networks (CDNs) facilitate an efficient, wide-area replication of static data for selected content providers, whereas the end-to-end design of TCP/IP does not foresee implicit replication and in-network storage. There is no openly available network standard for the asynchronous, global replication of popular content in the current Internet.

Inspired by the use case of widely deployed Content Delivery Networks (CDNs), current trends of *Information-Centric Networking (ICN)* shift the Internet towards data awareness. In ICN, consumers shall retrieve content by name directly from a network that provides storage, caching, content-based rendezvous, and searching at times. Thereby data sets become first class routable objects and content names require exposure to the control plane.

Several proposals have been presented in recent years [1], among them TRIAD [2], DONA [3], NDN [4, 5], PSIRP [6], and NetInf [7], which differ in several design choices. As we are interested in the stability and security of ICN infrastructures, we will concentrate on the aspects of routing and forwarding.

Essentially two approaches to routing exist in current ICN proposals, an evolutionary path that resolves names to locators and routes on IP (or a related location scheme), and 'clean slate' concepts that route directly on content names. NetInf extends the current Internet by a resolution service that maps content names to topological IDs like IP addresses, but alternatively supports name-based routing. TRIAD, DONA, and NDN perform content retrieval by routing on names. Route responses and the data itself are then forwarded along reverse paths (RPF), either by using IP as a lower layer, or without IP but by dedicated RPF states. PSIRP publishes content objects to a resolution system that incloses full knowledge of the network topology. Requesters trigger the mapping system to generate source routing identifiers in the form of Bloom filters that aggregate IDs of forwarding links.

All solutions operate on the content itself, and force the network infrastructure into a content awareness. A mapping service is not only required to resolve *file* names to source locations, but must answer a request by advising a nearby replica, the existence of which it learned from the data distribution system. Content routers need to rely on (often aggregated) names in its interface tables and—for RPF-based forwarding schemes—a reverse state for every data unit. This control information is highly dynamic and requires regular updates from the data plane. The ICN paradigm thereby opens up the control plane to continuous modifications from the data plane. This is in contrast to the current Internet, where DNS and routing states remain unaltered when a Web page is published, a file is transfered, or data is cached.

In this paper[1], we study the impact of traffic conditions on the control

[1] This paper is an extended version of [8], which includes simulation results and additional discussions. We started our work on this topic in the technical report [9].

plane. We are in particular interested in threats to the stability and security of the ICN infrastructure, whose impacts we evaluate in a theoretical analysis, experimental trials, and simulations based on real-world topologies. Experiments are performed in test networks running PARC's CCNx software. We want to stress, though, that our tests only attribute for the core concepts of content routing and do not evaluate implementation properties of the CCNx prototype. Following the basic insights gained from theoretical and practical analysis, we contribute a sample set of attacks that ground on this correlation of data with control states. We argue that the novelty of these exposures derives from an intrinsic binding to ICN concepts so that attacks—even if reminiscent from today's Internet—cannot be mitigated by simple protocol provisions.

The remainder of this paper is organized as follows: The specific problems in protecting the ICN infrastructure are stated in § 2 along with related work on ICN security. We theoretically analyze basic threats to stability in § 3 and discuss related implications. Based on practical experiments, threatening scenarios and their effects on the routing system are demonstrated in § 4. Correlation effects in stateful ICN routing are studied in simulations in § 5. These general insights lead to concrete attack scenarios in § 6. The paper concludes with a discussion in § 7.

2. Why ICN is Challenged by Design

2.1. ICN System Model

Information-centric networking involves two functional blocks within the network infrastructure, (1) content publications or announcements, and (2) content subscriptions or (asynchronous) access. Throughout this paper, we assume a generic ICN system model that is composed of these two subsystems, both of which introduce routing or forwarding states at the network layer. Even though not all ICN proposals are constructed equally pronounced in both parts, they all update corresponding table entries in response to data operations of the network infrastructure. In addition, we assume that universal caching is implemented in the content-centric routing system. Universal caching is common to all ICN solutions.

Content requests and delivery do not follow an end-to-end design, but require a dynamic set-up of paths between the requester and a (nearby) copy of the data. Commonly, this is done by Reverse Path Forwarding (RPF), at which each content request triggers a trail of 'bread crumb' states on routers along the path (NDN, DONA, NetInf). Alternative approaches that route on an underlying routing substrate like IP (NetInf), or constructs source routing identifiers based on complete knowledge of the topology (PSIRP), are not considered further in this work.

Some ICN implementations (e.g., CCN) signal error information when the content is not available. From this perspective they might be considered as request/response scheme. However, in the light of this article, publish/subscribe and request/response are not disjoint categories.

2.2. Problem Statement

Publishing and subscribing in current ICN solutions introduces network control states that generate the following management problems.

(1) Addressable content items need advertisement in the route resolution system. Consequently, any end user who can publish requires admission to modify the control plane.

(2) Content is conceptually delocalized by universal caching. Data replication thus imposes updates of the routing systems—a change of control state initiated by the data plane.

(3) Reverse Path Forwarding requires state initiation and consumption at routers along the path. Corresponding control state updates are not only driven by the data plane, but require processing in wire-speed.
These state operations raise the following threat classes in ways that are unique to ICN.

Resource Exhaustion Infrastructural entities need to offer accumulating resources like memory and processing power for provisioning, maintaining and exchanging content states. They are therefore threatened by resource exhaustion due to misuse or uncontrolled load. In addition, the asymmetry in size between data requests and delivery leads to traffic amplification when exploited in DoS attacks.

State Decorrelation The asynchronous nature of publish/subscribe content delivery places the enhanced burden of assuring consistency among distributed data states. Data states that require correlation are situated in distributed mapping systems, which also need to consistently reflect actual content placements, and in forwarding states at routers that define the paths hop-by-hop from a supplier to the requester. Failures in state coherence lead to service disruptions or unwanted traffic flows.

Path & Name Infiltration The infrastructure relies on the integrity and correctness of content routing and is therefore threatened by poisonous injections of paths and names, in particular. The replicative ICN environment distributes content copies to many, commonly untrusted locations and thereby makes it particularly hard to authenticate valid origins of state insertion requests.

All of these threats bear the potential to seriously degrade the ICN service and lead to insufficient or erroneous data dissemination. A major risk for the ICN infrastructure—and from a general perspective for the ICN concept—results from the power that an end user gains over an ICN distribution backbone.

2.3. Related work

Content suppliers Related work on ICN security has primarily focused on validating content correctness and authenticity. Commonly, self-certifying security credentials are included in 'secure names' that facilitate mechanisms for verifying authors, origins, and content integrity [10, 11, 12, 13]. Thus a receiver can be sure to obtain the correct content and an intermediate cache can validate the correctness of the security credentials, which prevents traditional DoS on the ICN system [14]. Nevertheless, having created (or learned) a valid name, any ICN member can re-announce this in the route resolution service, thereby injecting poisonous routes or artificial names into the system.[2] Similar vulnerabilities of DNS and BGP are known from today's Internet infrastructure [15], but remain restricted to (topology) *providers*. ICN opens the liberty of route injection to every content supplier. In an open Internet model, this can be any *end user*. We will discuss threats unique to ICN in Section 3.1.

Content consumers Little attention has been given to the effects of state management in ICN. Arianfar *et al.* [16] discuss design choices for an ICN router. They concentrate on the content cache and explicitly do not consider per request states. Perino and Varvello [17] have evaluated requirements for content routers that hold content information bases in Bloom filters and reverse paths in pending interest tables (PITs). Under the assumptions of *valid* content requests propagated on *homogeneous* network links with a *maximum global* RTT of 80 ms, average PIT sizes are identified in the order of 1 Gbit/s for current line speeds. FIB sizes and lookup complexity were shown to depend nonlinearly on prefix numbers and name lengths. Lauinger [18] explicitly addresses the threat of DoS attacks by filling the available memory of a router with pending interest states.

Such attacks on hardware resources may be mitigated by limiting overall table sizes. However, securing router resources by table limits does degrade network utilization and cannot abandon resource exhaustion problems. In the presence of a table limit, an attacker could initiate massive drops of pending Interests from a router's table and thus disrupt data delivery to regular receivers. The author in [18] proposes to drop Interests at the head of the PIT, which however may easily be misused to DoS-attacking neighbors, or to use Bloom filters instead of PIs. If applied without strict capacity limits, the latter approach is vulnerable to flooding attacks as interface filters degrade their selectivity. In the following section, we will evaluate these effects in detail.

Request state management and related security issues have been recently raised in [19, 20, 21]. Gasti *et al.* [22] address core issues of route hijacking, state overload, and cache pollution in NDN. They propose counter measures by extending interface functions, e.g., for limiting rates and survey content delivery. Without considering protective measures in BGP, the authors compare BGP with NDN security and argue that the NDN approach reduces vulnerability to black-holing, as routers can identify unresolved content requests and rank/re-route per prefix and interface. Authors miss that on the one hand RPKI secures BGP against hijacking attacks in a straight-forward manner, while on the other hand proposed countermeasures in ICN cannot prevent attacks of interception and redirection with service degradation.

Intermediate Summary ICN opens the control plane of backbone routers for content consumers and suppliers on a fine-grained base. Grant-

[2] As a countermeasure, DONA introduces certificates of publishers on the price of per cache-instance varying names. Content routing then works on wildcarding names, which re-introduces the threat of route poisoning.

ing end users access to the routing and forwarding subsystems is a fundamental step away from the current Internet design and bears significant risks. Current concerns in the context of routing mainly focus on state explosion due to the large amount of content items. One might argue that those resource exhaustions will be solved by more powerful hardware in the future. We will discuss options and limitations of related core aspects in Section 3. Still, binding the integrity of the routing infrastructure to the courtesy of *all users* is intrinsic to current ICN approaches—and presumably to the overall ICN concept.

3. Basic Threats to Stability

In this section, we theoretically examine the implications at the control plane for the different data operations and discuss resulting threats that inherently arise at the infrastructure level.

3.1. Routing or Mapping Resources

The common view on routing is that of a topological resolution service: Routing guides the paths to hosts. As ICN abandons the host-centric paradigm to address content objects directly, routes to content items attain the role of traditional topological directives.

State and Update Complexity In ICN, each content item (file) needs retrieval and therefore must be accessible via some resolution service. This may either be implemented by a distributed routing system, or by a mapping service that provides an indirection to topological locators of publishers or content caches. Whenever off-path caching is enabled, the average complexity of the corresponding management operations reads $\langle \#\ of\ content\ items \rangle \cdot \langle \#\ of\ cached\ replica \rangle \cdot \langle update\ frequency \rangle$ ($\langle \cdot \rangle$ denotes average values) and must be considered a severe challenge.[3] Solutions that are restricted to on-path caching reduce this complexity to $\langle \#of\ content\ items \rangle \cdot \langle update\ frequency \rangle$. In both cases, the request routing/mapping system is stressed by adding and updating name or – if applicable – cache entries at high frequency, the details of which depend on the implementation of the service.

Cache Announcements Route maintenance in ICN consists of propagating content publishers (i.e., default paths) as well as cache instances. While the first task is known to generate a high volume of data and frequent updates, caching is expected to largely exceed default announcements in number and update frequency. As a countermeasure, data replication may be limited to caching along default paths, which remarkably reduces the complexity for the routing system. On-path cache replica are met implicitly when requests are routed towards the source. They need not be advertised in the routing or

[3] A global request routing system will need to host at least the amount of the Google index base ($\mathcal{O}(10^{12})$) at a much enhanced update frequency (for timely content access and caching). For comparison, today's DNS subsumes $\mathcal{O}(10^8)$ names at a very low change rate of $\approx 10^5$ alterations per day.

mapping service. On the downside, restricting the caching to default paths will drastically reduce its effectiveness, and a corresponding strategy falls behind today's CDN solutions. Godsi et al. [14] discussed the caching problems in detail. The authors came to the conclusion that on-path caching is merely a warm-up of traditional web proxies.

Route Integrity ICN, like the current Internet, relies on the integrity of its routing system. A bogus route may block or degrade services, lead to incorrect content delivery, or violate privacy. These core concerns are well-known from BGP [15], where effective countermeasures exist. However, in addition to those vulnerabilities known from BGP routing, threats uniquely arise from data-driven state management in content-centric routing.

The first issue is inherited from universal caching. An explicit authorization of caches as common in the CDN market is in conflict with open publication and not applicable in general ICN approaches. Rather any node in the network can cache and thus announce any (forged) name, while origin validation measures such as RPKI [23] or [24] cannot be applied. The second issue emerges directly from state maintenance at routers. As the routing infrastructure is vulnerable to increased delays and delay variations in content supply (see Section 3.2), route redirections may be applied to slow down content delivery or to jitter response times. Following the first argument, any intermediate cache can—purposefully or accidentally—threaten its neighborhood.

3.2. Forwarding Resources

Traditional routers in the Internet consist of a central processing unit and main memory that are available to the control plane, mainly to learn and determine new routes, as well as FIB memory that is fed by the route selection process. Data forwarding remains bound to FIB lookup and packet processing at line-cards. This design choice purposefully decouples forwarding capacities from control processing and—with equal importance—protects control states from (bogus) data packets.

Current concepts of content-centric data forwarding break with this separation paradigm, and introduce—similar to IP multicast—an additional reverse path forwarding table, also called PIT. Unlike in multicast, this table is updated *packet-wise* on line speed by data-driven events. In the following subsections, we concentrate on the consequences for routing resources in detail. We will consider a chain of routers R_i along a data path and use the notation summarized in Table 1.

3.2.1. Content Request States versus Content Request Rates versus Network Utilization

Content request states are the essential building block to control flows in a content-centric distribution system that operates hop-by-hop. Each request state will trigger a data packet on return, why the number of open request states corresponds to data arrival at this interface after the transmission time.

Consider a point-to-point interface at routers R_i in steady operation and in the presence of a (per interface) state timeout T_i. In the absence of request

R_i	The i-th Router
C_i	Capacity of the link between R_i and R_{i+1}
U_i	Utilization of the link between R_i and R_{i+1}
S_i	# of content request states of R_i at its interface towards R_{i+1}
α_i	Content request rate at interface $R_i \rightarrow R_{i+1}$
ω_i	Content arrival rate at interface $R_i \leftarrow R_{i+1}$
T_i	Request timeout at interface $R_i \rightarrow R_{i+1}$
l	Packet length
$\langle \cdot \rangle$	Average value of $\cdot$
$\sigma(\cdot)$	Standard deviation of $\cdot$

Table 1: Glossary of Notations

retransmissions, packet loss, and state dismissal, we first want to derive the relation between routing request states at time t and network utilization. The total amount of state increases linearly by newly arriving requests α_i and decreases by content arrivals ω_i. Hence, the basic rate equation reads

$$\begin{aligned} S_i(t) &= S_i(t-T_i) + \int_{t-T_i}^{t} \alpha_i(\tau) - \omega_i(\tau) d\tau \\ &= S_i(t-T_i) + \int_{t-T_i}^{t} \alpha_i(\tau) - \alpha_i(\pi(\tau)) d\tau \\ &= \langle \alpha_i \rangle \cdot \min(\langle RTT \rangle, T_i) + \\ &\qquad \mathcal{O}\left(\sigma(\alpha_i) \cdot \sigma(\min(RTT, T_i))\right), \qquad (1) \end{aligned}$$

where $\pi(\cdot)$ denotes the time delay of the packet arrival process and RTT the random variable of packet round trip times, which is assumed independent of the requests and packet rates.

From Equation (1), we can immediately deduce that timeout values below the (varying) $RTTs$ limit the number of request states, but at the same time will block data forwarding. A second view reveals the strong dependence of routing state on the RTT variation. A similar phenomenon is well-known from TCP [25], but has been overlooked in corresponding previous work on ICN resource considerations [17, 21, 22].

Henceforth we will address the case of data flowing unhindered by the state timeout T_i and assume T_i large enough. Furthermore—for a steady-state scenario—it is assumed that the content request rate fluctuates on a stationary scale. Equation (1) then simplifies to

$$\begin{aligned} S_i(t) &\approx \langle \alpha_i \rangle \cdot (\langle RTT \rangle + \kappa\, \sigma(RTT)) \qquad (2) \\ &\approx U_i(t)/\langle l \rangle \left(\langle RTT \rangle + \kappa\, \sigma(RTT)\right), \qquad (3) \end{aligned}$$

with an estimating parameter κ for the mean deviation. The well-known term

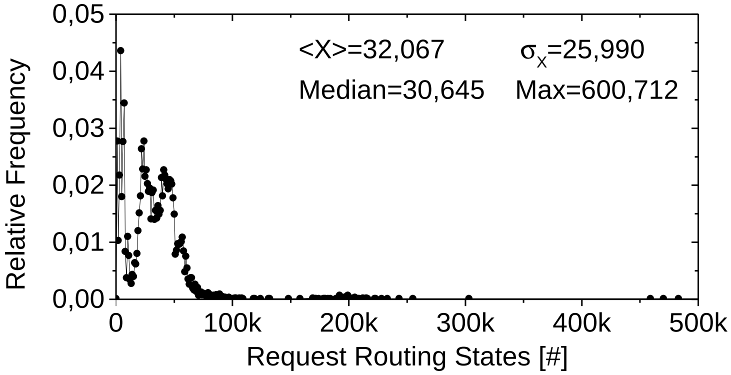


Figure 1: Distribution of forwarding states at routers with a 1 Gbit/s link, covering global RTTs [26] in March 2012

$(\langle RTT\rangle + \kappa\,\sigma(RTT))$ represents a retransmission timeout.[4] For the last step, we roughly assumed that content requests and content arrival are in stationary equilibrium.

Approximation (3) yields the desired coupling of the link utilization U_i and the state management resources at a router: On a single point-to-point link without state retransmissions and in flow balance, state requirements are proportional to the network utilization, enhanced by a factor of a *global retransmission timeout.* At switched interconnects or in bursty communication scenarios, conditions are expected to grow much worse.
The following observations are noteworthy.

1. Unlike in TCP that estimates a single end-to-end connection, content request states at routers subsume various prefixes and numerous flows. Moreover, content items (prefixes) are explicitly not bound to end points. Thus rapidly varying RTTs are characteristic to interfaces and even to individual flows in content-centric routing. The presence of chunk caching may further increase the RTT variation. Hence, no convergent estimator for a round trip time can be reasonably given.
2. In the current Internet, the variation of RTT is commonly larger than its average. End-to-end delays are known to approximately follow a heavy-tailed Gamma distribution [27]. PingER [26] reports means and standard deviations of about 250 ms, with maxima up to 5,000 ms. For a constant content request rate of 125k packets/s these RTTs generate the state distribution visualized in Figure 1.
3. Limiting the absolute size of the content request table imposes a strict bound on network utilization. However, the sustained rates are mainly

[4] The corresponding (over-)estimator in TCP is commonly set to 4. However, it is well known that standard TCP algorithms and parameters are inefficient at rapidly changing round trip times, which are characteristic for interface conditions in content-centric routing.

determined by actual RTTs and are hardly predictable. Similar arguments hold for defining timeout values.

4. Applying rate limits to content requests does not change the picture. For an 'on average' optimal limit $C_i \cdot \langle RTT \rangle / \langle l \rangle$, the variation of content replies in time may lead to large over- and under-utilization of network resources that goes along with large fluctuations in request table sizes.

3.2.2. Memory Requirements

A content-centric router that is designed to fully utilize its link capacities, requires sufficient table space for content requests under varying network conditions. Equation (3) approximates the corresponding resources when applied to the maximum link capacity C_i. Using the conservative value of $\kappa = 4$ as for TCP, a packet length l = 1,000 bytes, and RTT values from PingER as cited in the previous section, we derive

$$S_i = 1,25\, s/8.000\, bit \cdot C_i \approx 1,6 \cdot 10^{-4} s/bit \cdot C_i \tag{4}$$

For a line-speed of 1 to 100 Gbit/s, 160k to 16,000k content request entries then need to be installed per interface at minimum. Due to the more accurate consideration of *RTT* variation terms, these findings differ from previous results [17, 21] by more than an order of magnitude. Still they are merely a rough *lower estimate*, as larger fluctuations of round trip times may significantly increase resource demands.

It is noteworthy that Equation (4) holds for any router in a content-centric Internet. Unlike today, where full BGP tables are only required at AS border routers, and interior devices operate on a very small routing table, ICN access routers already demand for a full table memory, the size of which is determined by its interface capacities. In practice, this significantly increases router costs, as any fast interface must co-locate a large block of fast memory.

3.2.3. CPU Load from Table Management

An ICN router maintains states according to user data requests. For any content request, it needs at line speed to (1) insert a state in its request table. On the arrival of any data packet, it needs to (2) search and (3) delete on success in the same table. In addition, a router has to (4) maintain timers of all (soft) states in its request table. To guarantee robustness, an implementation of the huge request table not only needs to perform dictionary operations very efficiently *on average* but also in *worst-case*. With today's hash table implementations in software or hardware this is impossible to achieve [28, 29].

4. Experiments on State-based Forwarding

In this section, we present the results of straight-forward experiments that show the outcome of the core threats as theoretically discussed in Section 3. In particular, we concentrate on system and performance implications of the

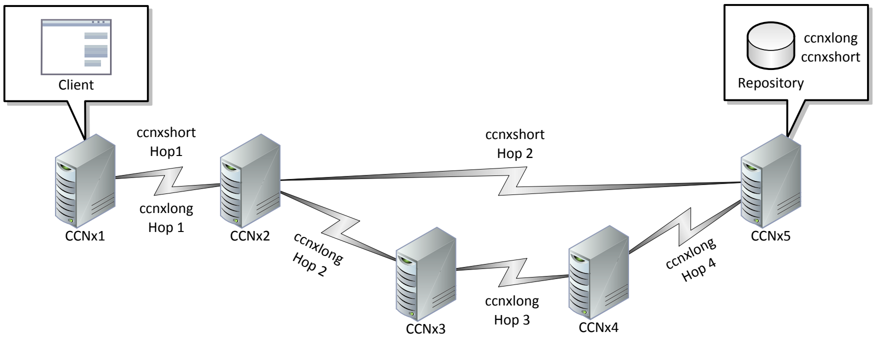


Figure 2: Topology of the experimental setting

data-driven state management at infrastructure devices. Even though the measurements mainly relate to the NDN implementation `ccnd`, we should emphasize that we do not evaluate the implementation itself, but use it as one real-world instance of the information-centric network deployment to illustrate the routing protocol mechanisms. Following this spirit, we do not interpret or discuss absolute performance values, which surely can be improved by optimized software and hardware in the future, but focus on structural and asymptotic analysis.

4.1. Core Measurement Setup

In our measurement study, we intentionally deploy *simple* communication scenarios between one content requester and one publisher. The basic network topology is represented by a Daisy chain of directly interlinked CCNx routers with 100 Mbit/s, one end connects the content consumer and the other the content repository (see Fig. 2). The *basic topology* consists of two hops and the *extended topology* of five nodes. It is noteworthy that more complex settings, e.g., a Dumbbell topology popular to represent backbone network effects, would enforce the effects, which we already see in our simpler and more transparent examples.

We use the CCNx implementation version 0.5.1 [30], i.e., the client library to announce content Interests, the content repository to store data, and the `ccnd` to forward subscription and data. The following analysis focuses on the effects on the router side. For obtaining a fine-grained view, we concentrate on the local system as well as inter-router dependencies.

We keep default values for all CCNx parameters. In particular, routers do not follow a specific strategy layer, as this would twist robustness towards specific limits as discussed in Section 3.2. CCNx routers communicate via TCP (preserving packet order in the basic experiments) or UDP (extended experiments).

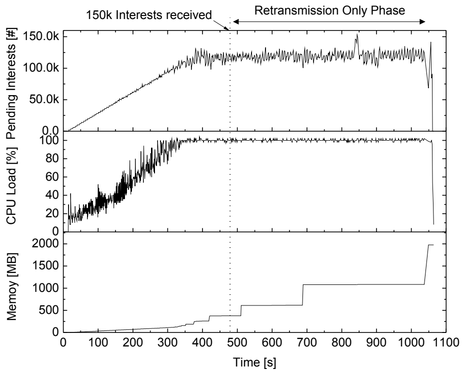


Figure 3: Load at the designated router of the receiver while requesting non-existing content

4.2. Basic Experiments: Resource Consumption

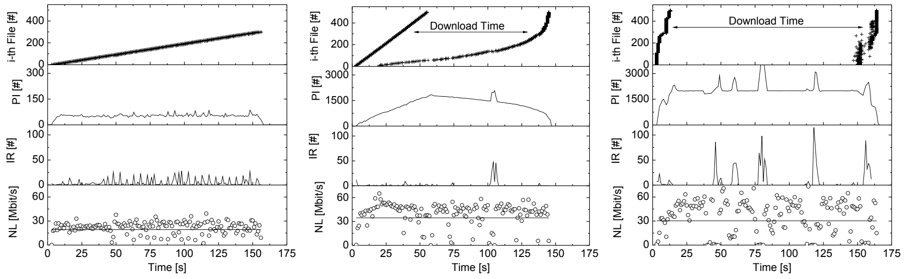
4.2.1. A Fast Path to Resource Exhaustion

An elementary threat intrinsic to data-driven state management arises from the overloading of routers by Interest requests (i.e., Interest flooding). This is most easily provoked by initiating requests for content that does *not* exist. In our scenario, the consumer issues 2,000 Interest messages for *non*-existing content, waits 6 seconds, and repeats these steps until overall 150,000 Interests have been sent.

Figure 3 shows the local resource consumptions on the first hop of the content receiver. The number of entries in the Pending Interest Table (PIT), the CPU load, and the required memory increase linearly with subsequent bulks of Interest messages until the system is saturated. In this case, the router reaches its limits of processing and memory resources when storing $\approx$120,000 PIT entries. While sending Interests, the initiating node retransmits previous announcements to keep states fresh at the router. Even though the retransmission timer is below the expiration timer and network delays are very short, the PIT size fluctuates as entries drop due to overloading. After all initial Interest messages have been distributed, the content consumer only retransmits subscriptions.

Our experiment illustrates several problems: A router may easily exhaust PIT space, when content arrives late or not at all. However, even if it was able to store all entries, it would suffer from a 'retransmission only' phase. The retransmissions agglomerate over time and create a continuous stream of signaling that consumes CPU cycles. When the update rate is higher than the processing capabilities permit, retransmissions require buffering, which leads to additional memory overhead (cf., Figure 3). A high system load increases the probability of dropping a PIT entry even if its refresh message has been signaled in time. This again causes additional refreshs of the PIT data structure (add/delete calls) and fosters load.

In a recent publication, Yi *et al.* [31] propose to mitigate this threat by signaling content unavailability back to the original requester. Such `NACK` will cure



(a) 2 files per second

(b) 10 files per second

(c) 100 files per second

Figure 4: Parallel download of 10 Mbit files: Start and stop time of the download per file at the receiver & resource consumption at its designated router [Pending Interests (PI), Interest Retransmits (IR), and Network Load (NL) including the mean goodput]

the Interest retransmission effects discussed above for truly unavailable content. However, this workaround has limited effect, as NACK suppression introduces a new attack vector at the content supplier side, while a bogus requester can still harm the routing infrastructure (in particular its designated router) by iterating Interest messages over various names of unavailable content.

4.2.2. Chunk-based State Multiplication

To analyze the performance of content consumption, we conduct a bulk file transfer. At this, the content receiver initiates the parallel download of multiple 10 Mbit files over a constant time. We consider three scenarios, the request of 2 files, 10 files, and 100 files per second, which correspond to an underutilized, a fully loaded, and an overloaded link. Figure 4 shows the start and completion time of the download per file (top graph), as well as the PIT size, the effective number of Interest retransmissions, and the traffic load including the mean goodput at the first hop. For visibility reasons, we rescaled the y-axis of PI in Figure 4(a).

With an increasing number of parallel downloads, not only the download times increase significantly, but also the interval of the request and receive phase grows in the scenarios of (over-)load. While the download time is almost constant for two files per second (cf., Fig. 4(a)), the time-to-completion grows non-linearly for the downloads in cases of excessive parallelism (cf., Fig. 4(b),(c)). 150 s are needed to download *each* single file in the worst case (Fig. 4(c)), while the link capacity would permit to retrieve *all* files in about 10 s.

The reason for this performance flaw is visualized in the subjacent graphs. A higher download frequency leads to an increasing number of simultaneous PIT entries, which require coordination with the data plane. Each file request will be split into requests of multiple chunks, in which the generation of corresponding Interest messages will be pipelined. In contrast to Section 4.2.1, content exists. As soon as the content traverses, Interest states dissolve and thus release memory. These operations cause a simultaneous burst in CPU load (not shown) and result in growing Interest retransmits after droppings or timeouts (shown in second lowest graphs). This also leads to retransmissions of data chunks. As an overall net effect, the network utilization fluctuates significantly, but does not adapt to actual user demands: Even though data requests could fill the links easily, the average load remains about constant at 30 % of the total network capacity.

In this example we demonstrated that insufficient processing and memory resources will strictly prevent a proper link utilization. This problem cannot be mitigated by rate limiting, as reduced Interest transmission rates will simultaneously reduce network utilization even further (see Section 3.2.1).[5]

The only visible way to assure proper utilization of network resources requires appropriate routing resources, i.e., a PI table implementation that is sufficiently large and reliably operates at line speed. As we learned from the analysis in Section 3.2, corresponding solutions are not available today. At the current state of the art, an attacker can always reproduce the performance degradations by either blowing up RTT and its variation, or by injecting states that degrade the performance of the PI hash table of the routers.

4.3. Extended Experiments: State Propagation and Correlation

In our extended experiments, we take a closer look at hop-by-hop routing performance using the five node routing chain displayed in the lower part of Figure 2. Intermediate nodes are numbered from the designated router of the content receiver (first hop) to the router of the content repository (fifth hop). In the following three experiments, we specifically concentrate on correlation effects of the routing resources by controlling the environment using parametrizable virtual machines.

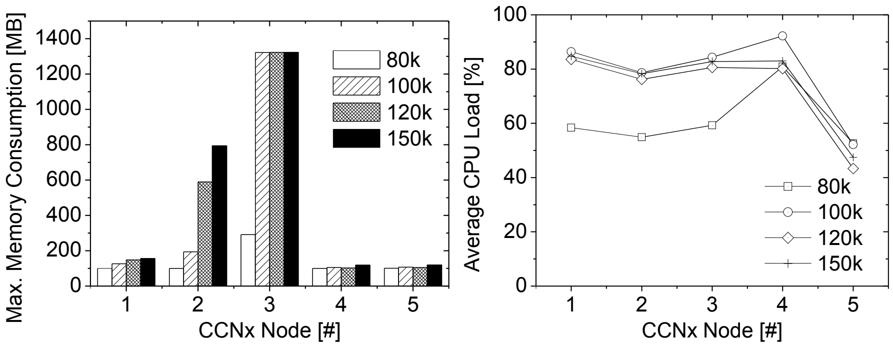
4.3.1. A Homogeneous Network

In this first extended experiment, we simply move our previous picture to the larger topology. All forwarding nodes offer the same resources, two cores@2.4 GHz, 3 GB RAM, and link capacities of 100 Mbit/s. A content requester downloads 500 files of size 10 Mbit at an average rate of 100 files per second. We observe a flattening of Interest propagation towards the source, as states resolve earlier from faster packet delivery (cf., Fig. 8(a)).

4.3.2. A Single Point of Weakness

It is a valid assumption that the content distribution system will consist of heterogeneous devices in terms of all performance metrics. In this second experiment, we introduce device heterogeneity by weakening a single router, the 4th hop (CCNx4), in a controlled way. We want to study the reaction of state management and network performance to this well-defined degradation.

[5] We should remind that applying Interest rates in NDN is a mechanism of flow control, and *not* for system resource protection. Intermingling these two aspects is likely to produce unwanted performance flaws and leads to new attacks (cf., Section 6).



(a) Memory Consumption

(b) Average CPU Load

Figure 5: Load per hop for a chain of 5 routers while initiating a 80k, 100k, 120k, and 150k different Interests for non-existing content

For an initial observation of the dependency on the weakest node, we reduce the CPU capacity of CCNx4 to 25 % (600 MHz) and recap the scenario from Section 4.2.1 for 80k, 100k, 120k, and 150k subscriptions of non-existing content. Independent of the capacity of the network infrastructure, the consumer initiates content subscriptions and continuously refreshes its Interests, which then propagate towards the content repository.

Figure 5 shows the maximal memory consumption and the average CPU load per hop during the measurement period. It is clearly visible that the required memory mainly depends on the position of the node within the topology. Memory requirements on the single path fluctuate by two orders of magnitude. The predecessor of the node with the lowest processing capacities (i.e., the 3rd hop) needs 50% – 500% more memory than any other nodes.

We now take a closer look on gradual effects of routing heterogeneity. We observe corrective mechanisms of the network (i.e., Interest retransmissions) depending on router asymmetry. Interest retransmissions serve as the key indicator for timeouts due to router overload. For this task, we configure CCNx4 with four different processing capacities related to the other CCNx routers: 2,400 MHz (homogeneous), 1,200 MHz (50 % capacity), and 600 MHz (25 % capacity).

Surprising results are shown in Figure 6. Evidently we see an instability in the forwarding behaviour of the network. The characteristic picture of a balanced network is a steady decay of Interest retransmits towards the source, as data delivery gets faster and more reliable in proximity to the publisher. However, at the first occasion of a 'bottleneck'—independent of its strength—the picture flips. Interest retransmission drastically increases and all routers except for the bottleneck equally see about the maximal rate of retransmissions in this scenario. State retransmissions at the weak forwarder (CCNx4) instantaneously doubles to the maximal level of managed states this router can cope with.

This experiment clearly shows how sensitive content-centric routing reacts to

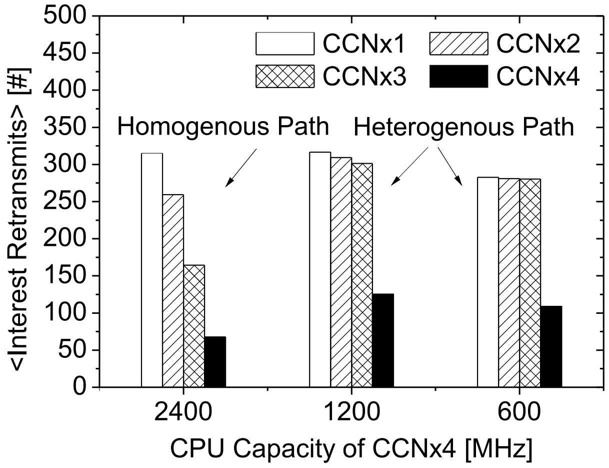
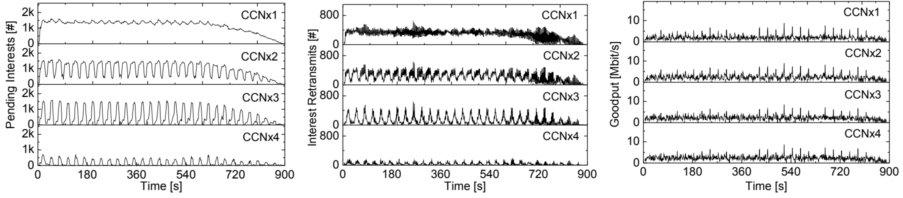


Figure 6: Effect of routing heterogeneity on Interest trading



(a) Pending Interests

(b) Interest Retransmits

(c) Network Utilization

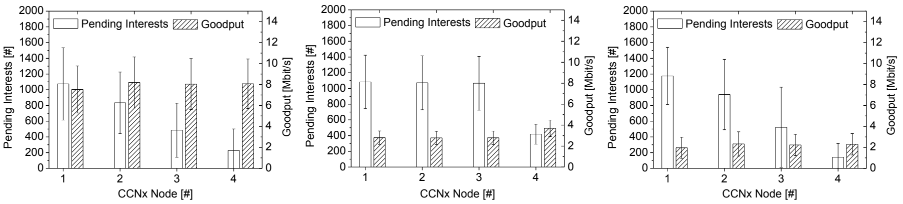
Figure 7: Routing and forwarding performance in a five-hop network with alternating CPU reductions

varying network resources. A light disturbance of the state propagation process reveals the instability of a steady-state flow by immediately turning content transport into a significantly different condition of maximal error management.

4.3.3. Complex Inhomogeneities

In our final experiment concerned with content routing, we explore situations of largely decorrelated network conditions. Therefore we configure all routers to admit fast changing resources occurring in anti-cycles. In detail, each router (CCNx1, . . . , CCNx5) is forced into a 10 s periodic CPU reduction by 90 %. Resource reduction periods were shifted between routers at a rate of 10 s so that at least one of the three routers in the forwarding chain was kept in challenged conditions. The objective of this repelling setup, which similarly may well occur from different side traffics in a meshed backbone, is to analyse the vulnerability of hop-by-hop state maintenance in ICN routing.

Results, i.e., Pending Interests, retransmits, and network throughput of this alternating resource scenario are displayed in Figure 7. The course of pending Interests as well as Interest retransmissions open a distinguished view on the fine-grained sensitivity of content routing to neighboring router conditions.



(a) Homogeneous Network (b) Single Point of Weakness (c) Alternating Resources

Figure 8: Comparison of state management and forwarding performance in different network scenarios (mean and standard variation)

State provisioning fluctuates on the resource resolution scale of 30 s throughout the network. More importantly, data transmission rates drop down to about 2.4 Mbit/s, while the overall load of Interest states remains compatible to the homogeneous network. Uncoordinated network resource availability thus leads to a low overall performance in conjunction with high network resource consumptions. Time-to-completion for each file download correspondingly explodes to 900 s for the same 10 Mbit files as in our initial experiment. It should be recalled that network capacities do allow for a simultaneous download of all 500 files within 10 s.

A comparative result of the different scenarios in our experimentally-driven analysis is presented in Figure 8. We contrast the load imposed onto the infrastructure by Interest states with the average network performance in the three experimental scenarios, homogeneous network, single point of weakness, and alternating resources at routers. The striking picture in all three settings is that the efficiency of network utilization is low on the overall, but drastically drops whenever inhomogeneities occur. The hop-by-hop forwarding performance thus appears rather fragile. In contrast, network state propagation attains various patterns, but always remains at compatible level at the router of maximal load.

These observations suggest the following rule of thumb for CCN routing performance: State maintenance always follows the maximal requirements, while forwarding performance will adapt to the weakest resource in place. This overall picture is clearly inefficient and future work on ICN solutions would largely benefit from improving this behaviour.

5. Simulation of Complex Networks

In our previous experimental evaluation, we have concentrated on simple topologies and on an in-depth analysis of individual router behaviour under data-driven state management. We will now focus on the overall performance of complex networks built from real-world topologies that we import into discrete event simulations. It is noteworthy that discrete event simulations do not experience load when managing states, but solely account for the interplay of request-routing and forwarding in the overall networking system.

5.1. Simulation Setup

Network simulations are based on ndnSIM [32], release 6 Nov. '12, an NDN implementation for NS-3. Our reference topology is built from the Rocketfuel [33] data set, which is commonly used in the ICN context [34, 35]. In detail, we started from the Sprintlink topology (# 1239) as core network of 315 nodes. These core routers are interconnected by point-to-point links with latencies obtained from the data set and homogeneous bandwidths of 10 Mbit/s. This backbone topology is extended by adding three additional edge nodes at each core router. The connections between core routers and associated edge nodes are via direct links of 1 Mbit/s with a latency of 10 ms. We should emphasize that bandwidths have been assigned with the aim of balancing the network. Absolute values carry no meaning, as we study effects of relative network performance.

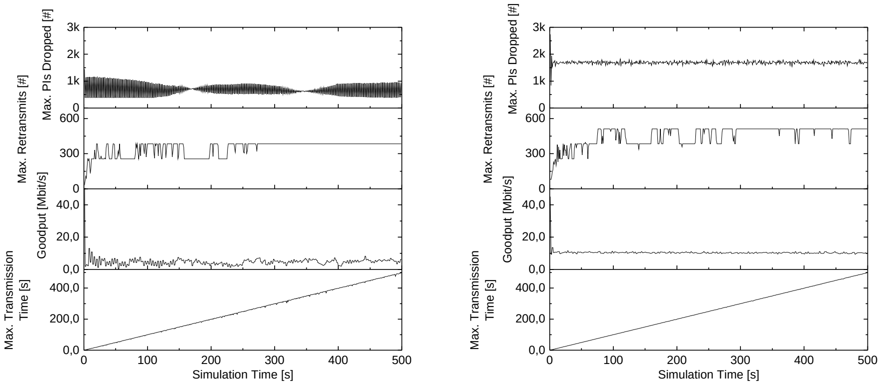
Every simulation node is provided with a protocol stack consisting of the link-layer Face *(ndn::NetDeviceFace)*, and the NDN protocol *(ndn::L3Protocol)* implementation. *ndn::BestRoute* implementation is used as Forwarding Strategy, whereas the Content Store module is not in use, and left uninstantiated in our configuration. For analyzing the coherence of states in the complex network, we limit the PIT sizes at all routers to 100. This value corresponds to a balanced utilization of the network core at 10 Mbit/s for the given average delay of 80 ms in our topology. We apply and compare the two PIT replacement strategies *persistent*, which keeps table entries and drops newly arriving requests on overload, and *random*, which randomly replaces entries of a full table.

Communication in the network is between *ndn::Producer* and *ndn::ConsumerCbr* applications. The consumer application issues Interests at a configurable frequency, and thus initiates data transfers. The producer applications are configured to reply with a data packet of 1024 Bytes in response to each arriving Interest that addresses a matching name. In each simulation run, we create 20 producers that are randomly placed either on edge or on core nodes. Regardless of its position, there is at most one producer per node. Consumers are always placed at edge nodes in a random fashion.

During simulation runs, we monitor PIT states, Interest retransmits, and data forwarding at each node. We extract the following metrics: The *maximal numbers of Interest drops and retransmits* per router, which serve as indicators of state decorrelation and network stress, the *overall goodput* jointly attained at all receivers, and the *maximal transfer time* taken over all data chunks that are delivered at a time of measurement.

5.2. Results

In our simulations, we study communication scenarios that are balanced at network edges, while backbone links—like in today's Internet—may be overbooked. Figure 9 displays the typical network performance for such a case: 20 producers are placed at core nodes, and 20 consumers per content source request 128 chunks per second for a simulation time of 500 seconds. These data requests saturate the access link capacities of 1 Mbit/s at each receiver and lead to immediate process of dropping and retransmitting Pending Interest at



(a) Persistent PIT Management

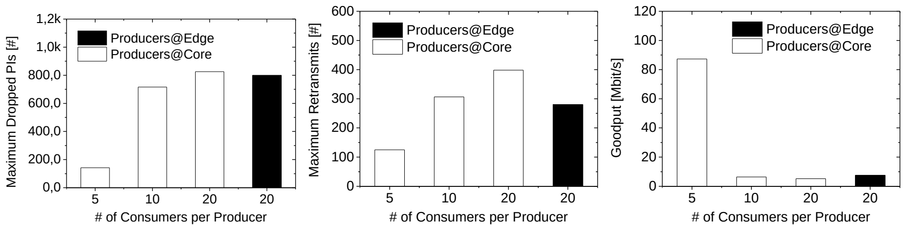
(b) Random PIT Management

Figure 9: Parallel chunk download: Stress and performance in a real-world topology

core nodes. It is clearly visible that dropping and retransmission frequencies are lower for the persistent PIT management (Fig. 9(a)), since request states that have entered routers on a complete path will remain present until data is forwarded. In contrast, the random replacement strategy may erase Interests on an established path and thus has a higher likelihood of decorrelating router states.

The overall forwarding capacity of the simulated scenarios is about 250 Mbit/s. Our simulation experiments both start out with a short peak in network utilization ($\approx$ 50 Mbit/s), but forwarding goodput quickly decays below 5 % of the capacity as soon as PIT overloads occur. Correspondingly, the slowest chunks arrive in times that grow linearly with the simulation time as a result of state decorrelation. Certain Interest states do not succeed in guiding a packet transfer until the backbone stress ceases, while timeouts and retransmits superimpose data forwarding in a mutually obstructive way. We observed the latter behaviour in all of our simulation runs with varying number of producers or request frequency. It is noteworthy that the persistent state strategy at routers avoids a dropping of data packets, since any path established from receiver to source remains intact for forwarding. As a consequence, the overall data goodput in the network corresponds to successful chunk deliveries, whereas a significant number of data packets ($\approx$ 0.5 per dropped Interest) is lost on path while the random replacement strategy is in operation. The enhanced goodput results in Fig. 9(b) are caused by undelivered packets and thus misleading.

Next we study the scalability and robustness of network performance with respect to varying capabilities and loads of producers for the case of persistent PIT management. We compare producers attached to the edge with producers in the core at increasing numbers of consumers. The results displayed in Figure 10 show a clear dependency of network integrity on request intensity. The more consumers request content services, the stronger the network decorrelates (see figures 10(a,b)). At the same time, the overall network performance drastically



(a) Interest Drops (b) Interest Retransmits (c) Network Goodput

Figure 10: Variation of producer position and number of consumers

decreases as visualized in Fig. 10(c). This comes at no surprise, as unsatisfied Pending Interests simply accumulate in the network with increasing probability of dropping on its path to the source. Placing producers at the edge hardly changes measurement results, even though the topological network balance is inverted. This confirms the observation that the network operations degrade due to state decorrelation along the path prior to arriving at the source.

Surprising in some parts, these simulation studies reveal a significant dependency of network performance and stress on the use patterns, independent of the specifics of the topology. Even though the discrete event simulations are robust with respect to system resources of router and—unlike in our experiments—performance remains unaffected by state maintenance, content-centric routing tends to not efficiently exploit transmission resources in realistic networks of intermediate size. Instead, our findings indicate that a state-wise uncoordinated hop-by-hop forwarding behaviour has unstable performance and can be easily degraded by adverse use patterns of consumers. This must be seen as a severe threat to the infrastructure, as a flooding of interests by end users can result in a denial of service attack at the remote core of the network.

Related phenomena of resource decorrelation are well known from Bittorrent-like systems [36], where a randomised resource trading often leads to a service degradation dominated by the weakest constituent. This is in contrast to the current Internet, whose paradigm of “Best Effort” actually defines a stable dynamic of maximized resource availability within the networking system. We suggest that future ICN research shall optimize the dynamics of network conditions under varying resource conditions.

6. Examples of Attack Scenarios

In this section, we briefly introduce attack scenarios for each threat enumerated in Section 2.2. Some attacks are unique to ICN, others—even though known from the Internet—gain a new level of severity by exploiting ICN intrinsics. We concentrate on attackers who exploit vulnerabilities of the infrastructure by generating various subscription resp. publication states as discussed in Sections 3 and 4.

6.1. Attacks Related to Resource Exhaustion

As shown in the previous section, routing and forwarding capacities of the infrastructure can be easily compromised by overloading its content request or interest tables. As non-aggregable name requests for locally unavailable content propagate through the network, resource exhaustion attacks can be transparently initiated from the remote. For this purpose it remains completely indifferent whether hardware resources are drained at unrestricted PIT sizes or table space exhausts according to various limiting configurations. Correspondingly, FIB overflows at routers occur in response to excessive publications or updates of names. Details of the attacker's effects depend on the state-dropping strategy—for simplicity we assume dropping tails as used by CCNx. In addition, virtual resources may also be depleted. The injection of bogus Interests disturbs ICN flow control mechanisms at routers, for example, because it reduces request limits.

Remotely Initiated Overload. An attacker that controls one or more machines (a botnet) may initiate massive requests for locally unavailable content based on Interest flooding (see Section 4.2.1). Corresponding interests propagate towards the publisher and eventually accumulate at some content router causing overload conditions. Depending on its intensity, this attack will lead to a service impairment or DoS for the (remote) content distribution tree(s) branching at the degraded router, unless the networking system is able to re-route the requests. It is worth noting that timeouts at regular users on the subtree will initiate retransmission 'storms' and thereby amplify the attack.

Piling Requests due to a Slow Source. Performance of a content source may be degraded by artificially high numbers of direct requests causing slowed down responsiveness. Alternatively, a captured source or its overloaded access router may drastically increase response times of content delivery. In slowing down a (popular) source, an attacker lowers the data return rate and thereby the extinction of pending interests at all routers on paths to receivers. Thus attacking a single point may result in a widely increased load at the routing infrastructure.

Mobile Blockade. A mobile node may issue a large number of invalid (or slow) Interests that block the state table of the access router for the period of state timeout. In a shared link-layer environment that cannot easily detect its departure, the mobile adversary can traverse neighboring networks on circular routes and continue to offload its interest bundle with the effect of a blockade of the regionally available networks. Initial countermeasures are difficult to apply, as the retransmission of Interests is part of the regular mobility pattern in ICN.

Fooling Rate Limiting. Current ICN approaches [21] propose rate limiting to restrict the number of Interest states. Its main purpose is flow control to avoid congestion. In contrast to common believes (see [22] for discussions) we argue that this is not an appropriate countermeasure to protect the ICN distribution system against attacks. An attacker can easily create an interest storm that exceeds the anticipated interest limit. The dedicated router will throttle the number of accepted interests per interface or interface+prefix, and finally ignore subsequent interests. Consequently, a single end user blocks a prefix or harms all members of its domain.

Note, applying rate limiting per end host (or user) is non-trivial in ICN. ICN explicitly discontinues the concept of host identifiers (e.g., due to security reasons). Thus, a router cannot track particular sources that send an unexpected amount of interests. Even if routers are enabled with such a function, an attacker can spoof addresses. The same holds for push-back mechanisms [22], which signal an overload towards the source and thus try to isolate an attacker. In addition, an attacker that would receive such a control message can ignore it.

6.2. Attacks Related to State Decorrelation

ICN requires consistent states during the request routing phase *and* the asynchronous content delivery. While bogus announcements or flapping of routes may introduce loops or increase the likelihood thereof, incoherent forwarding paths may result in partial content transmission that uses network resources without success in data delivery.

Heterogeneity Attack. An attacker that controls several machines (e.g., a botnet) may direct requests to accumulate at a specific router in the network and generate a point of performance degradation in the core. Heterogeneity will cause a significant service depletion for all crossing flows (see Section 4.3), if the network does not reroute. In the presence of rerouting, the adversary may use the same attack to trigger route flipping with corresponding jitter enhancements, which—in contrast to the Internet—will degrade access router performance for consumers.

Infringing Content States. An attacker that controls end systems or content routers could announce updates of content or cache appearances at a high frequency that exceeds the routing convergence time. As a consequence, the overloaded route resolution service will be unable to correctly process the updates of proper content sources or caches with the effect of incomplete content representation and erroneous data replication states. Content requesters will be thus led into false retrievals or access failures. As content announcement is commonly built on soft-state approaches, failures will timeout after a period of undisclosed inconsistency, which the adversary could initiate in a momentary attack.

Jamming Attack. A node on a shared link may issue a large number of content requests without maintaining the Interests at its own (loosing interest). Content will then arrive at the local link without a receiver. This is particularly harmful in mobile environments of limited bandwidth. A mobile attacker can jam a region by traversing shared radio links while requesting bulk data.

6.3. Attacks Related to Path and Name Infiltration

ICN raises content names and cache locations to first class objects and must therefore remain open to naming and placing data. The request routing system carries routes to names in its FIB or a mapping service, both of which are vulnerable to resource exhaustion and route poisoning. While an explosion in the pure number of names may be mitigated in parts by aggregation according to some authoritative naming conventions like in today's domain names, bogus route infiltration must be considered the more delicate issue.

Route-to-Death. An adversary that controls a cache system may redirect routes to it and slow down content delivery or jitter response times. As the routing infrastructure is vulnerable to increased delays and delay variations, resource exhaustion threats apply to the requesting infrastructure (see Section 3.2). In the presence of universal caching, reasonable counter measures to using a valid, but alienating cache are difficult to define.

Route Set Inflation. An adversary may announce bogus routes to cached copies of any content object. Content requests from its vicinity are then directed towards an erroneous location and—if unanswered or retarded— lead to long-lasting forwarding states and a possible DoS. This threat can be mitigated by resource-intensive attempts to route towards multiple locations that become increasingly painful when an attacker controls a botnet and injects invalid routes at large scale.

7. Discussions and Conclusions

In this paper, we have analyzed network instabilities and threats in information-centric networks that are caused by (a) backbone control states initiated by end users and (b) data-driven state management.

Some threats are easy to anticipate (e.g., resource exhaustion), others are more intricate due to the complex interplay of distributed management (e.g., state decorrelation). For the latter previous practical insights in the design of (conceptually related) multicast protocols already revealed good and bad design options. One of the major design goals of Bidirectional PIM [37], for example, was „eliminating the requirement for data-driven protocol events"— after the operating experiences with data-driven DVMRP or PIM-SM. With this paper, we want to stimulate the discussion about basic security in content-centric backbone routing.

Today, (D)DoS attacks are usually directed towards end hosts. In this paper, we have shown that ICN extends these threats to the backbone by design, and that existing countermeasures against both, DDoS and incorrect distribution states fail in the ICN field.

Defending from DDoS is already complicated in the Internet and becomes more intricate in ICN. From the conceptual perspective, the core challenge is not in deploying accountability (e.g., [38], [39]) but identifying an attack. Attack detection approaches [40] usually make application specific assumptions about traffic patterns, which cannot be applied to a generic Internet service for content delivery. We showed that the very fluctuating Internet delay space challenges resource provision in ICN (cf., Section 3.2). As content states will accumulate in the network (cf., Section 4), and inter-provider deployment almost surely will lead to a heterogeneous, unbalanced design, rate limiting may milden, but cannot effectively prevent the resource exhaustion problems discussed in this paper.

Current CDN deployments remain agnostic of these infringements by running under proprietary regimes. Present ICN proposals do not seem to have taken up the battle of standing in the wild. Very recently countermeasures have been proposed to mitigate Interest flooding attacks [41], [42], [43], [44]. In future work, we will analyze these extensions in more detail. In an open Internet, threats are built on the worst scenarios, not on average cases. If we want an information-centric Internet to remain open and reliable, a major redesign of its core architecture appears inevitable.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments. We gratefully acknowledge inspiring discussions with the community at the IRTF ICNRG meetings and the Dagstuhl Seminar 12361. This work was partially supported by the German BMBF within the projects H∀Mcast and Peeroskop.

References

[1] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, A Survey of Information-Centric Networking, IEEE Communications Magazine 50 (7) (2012) 26–36.

[2] M. Gritter, D. R. Cheriton, An Architecture for Content Routing Support in the Internet, in: Proc. USITS'01, USENIX Association, Berkeley, CA, USA, 2001, pp. 4–4.

[3] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, I. Stoica, A Data-Oriented (and beyond) Network Architecture, SIGCOMM Computer Communications Review 37 (4) (2007) 181–192.

[4] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, Named Data Networking (NDN) Project, Tech.report ndn-0001, NDN (2010).

[5] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, Networking Named Content, in: Proc. of the 5th Int. Conf. on emerging Networking EXperiments and Technologies (ACM CoNEXT'09), ACM, New York, NY, USA, 2009, pp. 1–12.

[6] P. Jokela, A. Zahemszky, C. E. Rothenberg, S. Arianfar, P. Nikander, LIPSIN: Line Speed Publish/Subscribe Inter-networking, in: Proc. of the ACM SIGCOMM 2009, ACM, New York, NY, USA, 2009, pp. 195–206.

[7] B. Ahlgren, et al., Second NetInf Architecture Description, Tech.report D-6.2 v2.0, 4Ward EU FP7 Project (2010).

[8] M. Wählisch, T. C. Schmidt, M. Vahlenkamp, Lessons from the Past: Why Data-driven States Harm Future Information-Centric Networking, in: Proc. of IFIP Networking, IEEE Press, Piscataway, NJ, USA, 2013.

[9] M. Wählisch, T. C. Schmidt, M. Vahlenkamp, Backscatter from the Data Plane — Threats to Stability and Security in Information-Centric Networking, Technical Report arXiv:1205.4778, Open Archive: arXiv.org (2012). URL `http://arxiv.org/abs/1205.4778`

[10] W. Wong, P. Nikander, Secure Naming in Information-centric Networks, in: Proc. of Re-Architecting the Internet Workshop (ReARCH '10), ACM, New York, NY, USA, 2010, pp. 12:1–12:6.

[11] C. Dannewitz, J. Golić, B. Ohlman, B. Ahlgren, Secure Naming for a Network of Information, in: Proc. of the IEEE Global Internet Symposium, IEEE, Piscataway, NJ, USA, 2010.

[12] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, S. Shenker, Naming in Content-oriented Architectures, in: Proceedings of the ACM SIGCOMM workshop on Information-centric networking, ICN '11, ACM, New York, NY, USA, 2011, pp. 1–6.

[13] N. Fotiou, G. F. Marias, G. C. Polyzos, Publish–Subscribe Internetworking Security Aspects, in: N. Blefari-Melazii, G. Bianchi, L. Salgarelli (Eds.), Trustworthy Internet, Springer, Heidelberg, 2011, pp. 3–15.

[14] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, J. Wilcox, Information-Centric networking: Seeing the Forest for the Trees, in: Proc. of the 10th ACM HotNets Workshop, HotNets-X, ACM, New York, NY, USA, 2011.

[15] K. Butler, T. Farley, P. McDaniel, J. Rexford, A Survey of BGP Security Issues and Solutions, Proc. of the IEEE 98 (1) (2010) 100–122.

[16] S. Arianfar, P. Nikander, J. Ott, On Content-Centric Router Design and Implications, in: Proc. of ReARCH workshop, ACM, New York, NY, USA, 2010.

[17] D. Perino, M. Varvello, A Reality Check for Content Centric Networking, in: Proc. of the ACM SIGCOMM WS on Information-centric Networking (ICN '11), ACM, New York, NY, USA, 2011, pp. 44–49.

[18] T. Lauinger, Security & Scalability of Content-Centric Networking, Master's thesis, TU Darmstadt, Darmstadt, Germany (2010).

[19] Y. Chung, Distributed Denial of Service is a Scalability Problem, ACM SIGCOMM CCR 42 (1) (2012) 69–71.

[20] M. Wählisch, T. C. Schmidt, M. Vahlenkamp, Bulk of Interest: Performance Measurement of Content-Centric Routing, in: Proc. of ACM SIGCOMM, Poster Session, ACM, New York, 2012, pp. 99–100.
URL http://conferences.sigcomm.org/sigcomm/2012/paper/sigcomm/p99.pdf

[21] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, L. Zhang, A Case for Stateful Forwarding Plane, Tech. Rep. NDN-0002, PARC (July 2012).

[22] P. Gasti, G. Tsudik, E. Uzun, L. Zhang, DoS and DDoS in Named-Data Networking, Tech. Rep. 1208.0952, ArXiv e-prints (August 2012).

[23] P. Mohapatra, J. Scudder, D. Ward, R. Bush, R. Austein, BGP Prefix Origin Validation, Internet-Draft – work in progress 10, IETF (October 2012).

[24] A. Li, X. Liu, X. Yang, Bootstrapping Accountability in the Internet We Have, in: Proc. of the 8th NSDI, USENIX Association, Berkeley, CA, USA, 2011.

[25] V. Jacobson, Congestion Avoidance and Control, SIGCOMM Comput. Commun. Rev. 18 (4) (1988) 314–329.

[26] PingER. Ping end–to–end reporting, http://www-iepm.slac.stanford.edu/pinger/ (2012).

[27] C. J. Bovy, H. T. Mertodimedjo, G. Hooghiemstra, H. Uijterwaal, P. V. Mieghem, Analysis of End to end Delay Measurements in Internet, in: Proc. of the Passive and Active Measurement Workshop-PAM, 2002.

[28] S. A. Crosby, D. S. Wallach, Denial of Service via Algorithmic Complexity Attacks, in: Proc. of USENIX Security Symposium, USENIX Assoc., Berkeley, CA, USA, 2003, pp. 29–44.

[29] U. Ben-Porat, A. Bremler-Barr, H. Levy, B. Plattner, On the Vulnerability of Hardware Hash Tables to Sophisticated Attacks, in: Proc. of IFIP Networking, Vol. 7289 of LNCS, Springer–Verlag, Berlin, Heidelberg, 2012, pp. 135–148.

[30] PARC, The CCNx Homepage, http://www.ccnx.org/ (2012).

[31] C. Yi, A. Afanasyev, L. Wang, B. Zhang, L. Zhang, Adaptive Forwarding in Named Data Networking, SIGCOMM Comput. Commun. Rev. 42 (3) (2012) 62–67.

[32] A. Afanasyev, I. Moiseenko, L. Zhang, ndnSIM: NDN simulator for NS-3, Technical Report NDN-0005, NDN (October 2012).
URL http://www.named-data.net/techreport/TR005-ndnsim.pdf

[33] R. Mahajan, N. Spring, D. Wetherall, T. Anderson, Inferring link weights using end-to-end measurements, in: Proc. of the 2nd ACM SIGCOMM Workshop on Internet measurment (IMW'02), IMW '02, ACM, 2002, pp. 231–236.

[34] J. Chen, M. Arumaithurai, X. Fu, K. K. Ramakrishnan, G-COPSS: A Content Centric Communication Infrastructure for Gaming Applications, in: Proc. of IEEE ICDCS, IEEE Computer Society, Los Alamitos, CA, USA, 2012, pp. 355–365.

[35] Z. Zhu, C. Bian, A. Afanasyev, V. Jacobson, L. Zhang, Chronos: Serverless Multi-User Chat Over NDN, Technical Report NDN-0008, NDN (Oct. 2012).

[36] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, X. Zhang, Measurements, Analysis, and Modeling of BitTorrent-like Systems, in: Pro. of 5th ACM SIGCOMM conference on Internet Measurement (IMC), USENIX Association, Berkeley, CA, USA, 2005, pp. 4–4.

[37] M. Handley, I. Kouvelas, T. Speakman, L. Vicisano, Bidirectional Protocol Independent Multicast (BIDIR-PIM), RFC 5015, IETF (October 2007).

[38] D. R. Simon, S. Agarwal, D. A. Maltz, AS-Based Accountability as a Cost-effective DDoS Defense, in: Proc. of Workshop on Hot Topics in Understanding Botnets, USENIX Association, Berkeley, CA, USA, 2007.

[39] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, S. Shenker, Accountable Internet Protocol (AIP), in: Proc. of the ACM SIGCOMM, ACM, New York, NY, USA, 2008, pp. 339–350.

[40] T. Peng, C. Leckie, K. Ramamohanarao, Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems, ACM Comput. Surv. 39 (1).

[41] H. Dai, Y. Wang, J. Fan, B. Liu, Mitigate DDoS Attacks in NDN by Interest Traceback, in: Proc. of IEEE INFOCOM NOMEN Workshop, IEEE Press, Piscataway, NJ, USA, 2013.

[42] A. Compagno, M. Conti, P. Gasti, G. Tsudik, Poseidon: Mitigating Interest Flooding DDoS Attacks in Named Data Networking, Tech. Rep. 1303.4823, ArXiv e-prints (March 2013).

[43] K. Wang, H. Zhou, H. Luo, J. Guan, Y. Qin, H. Zhang, Detecting and mitigating interest flooding attacks in content-centric network, Security and Communication Networks (2013) Early view online.

[44] A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, L. Zhang, Interest Flooding Attack and Countermeasures in Named Data Networking, in: Proc. of IFIP Networking, IEEE Press, Piscataway, NJ, USA, 2013.