T.C. Schmidt, S. Wölke, N. Berg, M. Wählisch, “Let’s Collect Names: How PANINI Limits FIB Tables in Name Based Routing”, in *Proc. of IFIP Networking Conference*, 2016.

Let’s Collect Names: How PANINI Limits FIB Tables in Name Based Routing

Thomas C. Schmidt, Sebastian Wölke, Nora Berg
Dept. Informatik, HAW Hamburg
firstname.lastname@haw-hamburg.de

Matthias Wählisch
Freie Universität Berlin
m.waehlisch@fu-berlin.de

***Abstract*—Name-based routing as proposed in Information Centric Networking encounters the problems of (a) exploding routing tables, as the number of names largely exceeds common routing resources, and (b) limited aggregation potentials, as names are commonly independent of content locations. In this paper, we introduce Partial Adaptive Name Information in ICN (PANINI), an approach to scale routing on names. PANINI aggregates names at (virtual) collectors and adapts FIB tables simultaneously to available resources and actual traffic patterns. PANINI introduces routing hierarchies and prefix-specific default routes, bimodal FIBs, and confined flooding. We thoroughly evaluate the approach in theory and practical experiments. Our findings indicate that effective reductions in control state largely outweigh overheads in control traffic.**

***Index Terms*—FIB aggregation, scalable adaptive forwarding, CCN/ NDN, confined Interest flooding.**

I. Introduction

Information Centric Networking has introduced a new, promising communication paradigm, but continues to struggle with severe challenges [1]. NDN [2] (among others) binds routing on names at a high level of maturity. However, the multitude and complexity of distributed content names has not been treated convincingly. Names are by orders of magnitude too many to be stored in today’s forwarding information basis (FIBs) and remain too delocalized to allow for aggregation. Even though several original approaches have been presented [3], [4], the sheer scalability demands risen from names prevent a striking step forward.

Scalable routing in the current Internet is achieved by a hierarchy that shields global from local operations. The majority of route identifiers is situated at the edge, but treated as aggregates at the core. The initial concepts of routing on hierarchical names invert this principle and require detailed, de-aggregated knowledge of name state throughout the network. In this paper, we approach this problem by introducing Partial Adaptive Name Information in ICN (PANINI).

The PANINI approach [5] starts from fixing an aggregation point for a group of names resident in a (topological) network. The typical aggregator would be a larger cache repository on the fixed Internet, or a gateway in the IoT. We assume topology building mechanisms in place that generate a shortest path tree rooted at the aggregation point. This is in full analogy to the current Internet, where standard routing protocols can construct shortest paths on the inter- and intra-domain level. Given this basic topology, every node can identify up- and downward paths with respect to the aggregating root—with upward paths serving as default.

ISBN 978-3-901882-83-8 © 2016 IFIP

The objective of name-based routing is to link content requesters with content suppliers in an efficient way. Inspired by the highly skewed popularity distribution of names, PANINI aspires to efficiently balance FIB sizes and control traffic. Popular names are included in distributed tables, while unpopular ones are omitted and searched by confined flooding. Our thorough evaluations reveal significant optimizations at small FIB tables *and* rare flooding events.

In the following, we will present this hybrid combination of (artificially enhanced) name aggregation at rendezvous points, adaptive mapping by FIBs, and a dynamic on-demand flooding of Interests towards content suppliers. We start with a problem statement and discuss related work in Section II. The PANINI routing and forwarding scheme is presented in Section III along with several deployment scenarios. Extensive evaluations accounting for both, theory and experimentation follows in Section IV. Finally, we conclude and give an outlook in Section V.

II. The Problem of Scalable Routing on Names and Related Work

A. FIB-size, Aggregation, Flooding

Scalable name-based routing is one of the open research challenges in ICN [1], [4], [6], [7]. This problem appears at least with two faces—limiting state (FIBs sizes) and control traffic at routers.

A common approach to reducing routing entries is aggregation. Aggregation of names, though, requires a correspondence of identifiers and locations. Such an assumption conflicts with a flexible or distributed content placement within the network. According to current common practice, content names belong to the content owner and not to the network provider. Consequently, a content owner can decide to change the ICN upstream, which then leads to de-aggregated routing entries. Furthermore, routers on names in ICN cannot locally decide on aggregation, since names—unlike IP addresses—do not fall into a fixed, enumerable number space.

The Internet consists (and will consist) of heterogeneous kinds of routers in terms of hardware and capacity. Assuming a flexible name-based content distribution system—as originally envisioned in the ICN community—routing tables will naturally aggregate rather sparsely and easily cause memory exhaustions at most routers. As long as a router is single-homed, all entries can be collapsed to a (default) entry towards the (single) uplink. Most mobile end devices as well as edge routers are multi-homed, though. In PANINI, we extend the concept of default routing and leverage its benefits without ignoring the potential of the underlying network structure.

An alternative approach to implement scalable routing is to separate names from locators and deploy a name resolution service [8], [9]. Staying with the core concepts of NDN/CCN and its security benefits, PANINI performs routing solely on names and without a mapping.

Converse to a complete table view or a default routing system operates a path detection by flooding. Flooding helps to explore the location of content but is clearly not applicable on Internet-scale. PANINI exploits flooding occasionally in strictly confined local regimes, when hardware resources are limited.

To reduce the amount of memory allocated by FIB entries, several data structures have been proposed that are specifically tailored to name-based routing (e.g., [10]). We consider those approaches orthogonal to PANINI, as they help to implement scalable name-based routing (wherever a complete view is required) but do not solve the scalability problem from first principle.

B. *Name-based Routing*

Recently, the debate on how to improve the state of name-based routing has heated again with several proposals. OCEAN [11] starts from the observation that aggregation is unlikely to occur on its own at Internet scale. Agreeing with this observation, we introduce aggregation facilitators. Instead, OCEAN proposes to aggregate on virtual paths and (re-)introduces a virtual circuit path switching facility. These 'pathlet'-type forwarding also eliminates loops that can occur in current NDN/CCN. By introducing a clear, Internet-type route hierarchy, loops are equally prevented in PANINI.

SNAMP [12] proposes a *map-encap* approach including mapping services at the edges, which link an arbitrary name to a backbone-specific prefix. Thus, routers in the default-free zone (DFZ) need to store only a subset of prefixes of the overall namespace but the edges need to handle full tables. SNAMP introduces inverse requirements as compared to PANINI where name collectors with complete FIB entries span the DFZ.

Wang *et al.* [13] designed a flooding mechanism for ICNs. Therein the flooding radius is set dynamically from local graph properties that yield information about the global structure. It shows that in scale free networks nearly all information can be retrieved within a very small flooding radius. In PANINI, we apply additional restrictions and can show that even in the worst case, the overhead which is introduced by flooding remains relatively moderate.

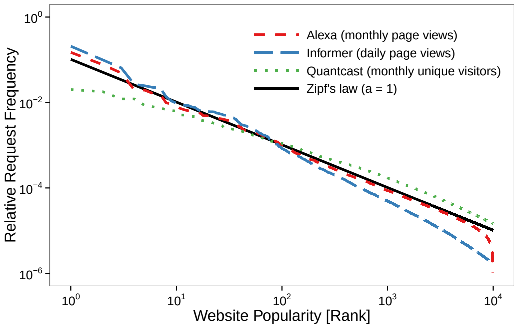


Figure 1. Zipf's law in comparison to empirical name popularity from different sources

Geometric routing [14]–[16] addresses the problem of route scalability by encoding forwarding in coordinates. These approaches are promising as they do not require a global routing table, neither in the edge nor in the core. However, embedding arbitrary names in geometric space is still an unsolved problem in real Internet-like deployment.

C. *Name Popularity*

The huge numbers of names for content can be contrasted by its largely uneven frequency of use, which PANINI exploits. The distribution of name popularity has been repeatedly measured in different contexts like web caching [17], or web access [18] and was found to be a power law distribution of Zipf type [19].

For confirmation and parameter fixing, we performed additional checks on web data of different type and periods. In detail, we consulted the three different web analytic services *Alexa*[1], which provides monthly page view statistics for the top million websites, *Quantcast*[2], which offers statistics of unique monthly website visits from the United States, and *Informer*[3], from which we crawled the daily visitors and page views per website.

Results are displayed in Figure 1 in comparison with Zipf's law for $a = 1$. All measurements remain in reasonable agreement with the Zipf curve, why we continue to build our content popularity model and analysis on it.

D. *Modeling Shortest Path Trees*

The routing mechanism of PANINI creates shortest path trees (SPTs). From theory [20], [21] we know that SPTs are well modelled by uniform recursive trees (URTs). URTs are random trees that can be generated stepwise as follows.

---

[1] http://www.alexa.com
[2] http://www.quantcast.com
[3] http://website.informer.com

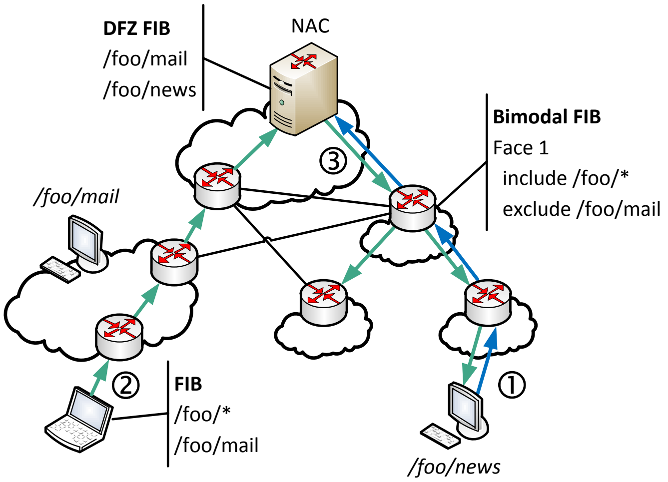


Figure 2. Overview of PANINI routing and forwarding.

First, the root vertex is added to an empty graph. In each following step n, the n-th vertex is connected to one of the $n-1$ previous vertices with equal probability [22]. The resulting tree is self-similar and represents the structural properties of shortest path trees within networks, as we will exploit in our analytical evaluation.

III. PANINI ROUTING

A. Overview

The core ideas of PANINI are to limit global as well as local FIBs by introducing aggregation points (*name collectors*) that are reachable via default routes, and an utility-adaptive FIB management.

1) Limiting Global FIBs: PANINI distributes responsibility for names across Name Collectors (NACs) by assigning prefixes to NACs. Each NAC aggregates those names that match its prefix(es) at a static preconfigured position in the network. Prefixes can be selected demand-wise and distributed among an arbitrary number of NACs, tailoring a partition of the routable namespace according to content popularity, topological preferences, or provider needs. We discuss diverse deployment scenarios in the following subsection.

As such, a NAC serves as a prefix-specific anchor point that aggregates name-based routes and facilitates name-specific caching. The anchor itself becomes reachable via prefix-specific default routes, which a NAC simply advertises within its domain. These default routes purely depend on the topology and remain independent of individual content providers. Following such a default route for a prefix, any node within the network can reach the corresponding NAC on the shortest available path. Hence the union of all default routes for a prefix will define a shortest path tree rooted at the NAC. Figure 2 displays such a default for the prefix `/foo/*`. From this perspective, NACs form the default-free zone (DFZ) in PANINI routing.

2) Publishing Content: A content supplier who wants to advertise a name to the routing system uses the default route towards its most specific prefix for issuing a Name Advertisement Message (NAM) (see step 1 in Fig. 2). Per default, NAMs travel hop-by-hop towards the aggregation point, and every intermediate router can harvest the content advertisement for including in its own routing table. These table entries are specific, down-tree oriented non-default routes. Filling all FIBs will generate a complete routing path from the aggregation point to the content source. It is worth noting that routing states close to a NAC naturally aggregate in FIBs.

3) Requesting Content: A consumer requests content by transmitting an Interest for a name. In the absence of more specific FIB entries or cache hits, this Interest will travel up to the NAC on the default route (see step 2 in Fig. 2), where popular content is likely to be cached. If not satisfied from the cache, the Interest will be forwarded down along the previously installed path to the content provider (see step 3 in Fig. 2). Data forwarding will follow the pending Interest states on the reverse path as in regular NDN/CCN. Routing and forwarding are thus aligned to a network hierarchy that resembles the current Internet with aggregation points located at the transit tier. It is noteworthy that routing towards the NAC will aggregate paths and thereby facilitate on-path caching.

4) Limiting Local FIBs: Up to this point, we have globally reduced FIB entries to prefix-specific defaults, but required names present in local FIBs. This is known to be infeasible in ICN. PANINI weakens this requirement as follows. Complete routing tables shall only be required at the aggregation points. This is a significant relaxation, since aggregation points are designed to facilitate name aggregation and largely reduce routing table space. In addition, providers may select strong NAC devices. From complete, aggregated FIB tables, the (transit) root can thus always tell which branch (or lower tier ISP) holds the requested content. Without further FIB entries, flooding may lead the Interest down this (loop-free) branch.

Intermediate nodes are not required to carry a full FIB, but rather aim at adapting selected entries to minimize Interest flooding. In analogy to caching content, each node autonomously decides about (a) its memory resources available for the FIB, and (b) the forwarding logic it applies within its vicinity. Traffic flows (with highly skewed name utilization) can be continuously used to adapt the FIB to relevant traffic patterns. For example, a node can hold more specific information for frequently requested names, while it may erase entries for traffic rarely seen.

To optimize Interest guidance with partial forwarding information even further, we introduce a *bimodal FIB*. This extends the FIB structure to operate in two modes—`include` and `exclude`. In `include` mode, all Interests that match a FIB prefix will be forwarded on the associate Face, while all Interests that match a FIB `exclude`-prefix will be blocked on that Face. The initial state of an empty FIB reads `exclude *` which prevents flooding of all incoming Interests. A node that has seen no routable names from NAMs about a prefix `/foo/*’` in a subtree of his will remain `exclude /foo/*`.

In combining these two routing mechanisms—(i) default prefix routes to NACs and (ii) adaptive bimodal FIB management—the PANINI system largely reduces FIB tables. In the evaluation Section IV, we will show that even with constant, small FIB sizes the routing remains highly efficient.

B. Deployment Scenarios

1) A Content Delivery Internet: In an Internet-wide deployment, PANINI content distribution will rely on per prefix replicated NACs that are placed in various provider networks. Each NAC serves its local content suppliers and requesters as the default addressee. Routing and forwarding for locally available or cached content remains local as described in the previous section.

To make content accessible across domains, NACs in service of the same prefixes need to peer in the default free zone. NACs need to exchange more specific prefixes of their local content names, which can be done similar to BGP network prefix exchange, or by a distributed key-value system. It is worth recalling that NACs are aware of their complete local name tables and can therefore efficiently aggregate. Request routing and content forwarding can then be performed either by directly traversing the local NAC and its peerings, or recursively by the local NAC as it is common in today’s CDNs. A detailed study of routing and forwarding in the DFZ will be subject to our future work.

2) ICN in the Internet of Things: IoT networks commonly consist of distributed sensors and actuators which are often constrained, embedded devices, and at least one (full-featured) Internet gateway. To serve this setting, we first need to create a topology. We propose to follow the well-established approach of building a tree-like structure—a destination-oriented directed acyclic graph (DODAG)—as known from RPL [23], for example. Parents broadcast their presence (DIO) and children attach (DAO). These link-local operations can be transfered to the link-layer in a straight-forward manner. The IoT gateway takes the role of the root node and NAC.

Given this basic topology, the gateway(s) can announce their default prefixes, which in a simple network will reduce to a unique default route (`/*`). In the IoT, we need to face the trade-off that Interests in a constrained environment should ideally be minimized, but intermediate nodes have limited memory and cannot hold large routing tables. In our previous work [24], we have designed and analysed two routing corner stones—Vanilla Interest Flooding (VIF) and Reactive Optimistic Name-based Routing (RONR). While VIF works without a FIB, RONR nodes gradually acquire all FIB entries in a reactive fashion. Given the DODAG topology, PANINI can now define a self-optimizing strategy for routing on names by making a hybrid use of both routing primitives—typically keeping the FIB limited to hold a few entries that are replaced according to a least frequently used (LFU) policy.

Table I
Example of a PANINI FIB table

Mode	Prefix	Face
`Default`	`/foo/*`	1
	`/bar/*`	1
`Include`	`/my/videos/`	2
	`/your/music/`	3
`Exclude`	`/qux/*`	2,3

3) Edge Caching in 5G Mobile Networks: The emerging 5G mobile network architecture foresees an ultra dense access network that is backed by a shared data domain for fast content access. Several major players including Cisco opt for deploying ICN technologies to facilitate content caching at the edge.

PANINI routing will significantly simplify deployment as follows. NACs shall be positioned as virtualized networked functions in the shared data domain to channel content retrieval and caching from the open Internet (either by a name-based peering or by traditional IP). NACs can dynamically resize in their virtualized environment to adapt caching capabilities to content popularity. They will distribute default content routes throughout the access network so that the ultra dense access only needs to carry a minimized set of FIB entries that persist with topology. Neither complex, user-driven route management nor flooding are required, as content is always pulled from the data sharing domain.

This setting resembles the base PMIPv6 multicast architecture [25], which experienced deployment. Here access routers (MAGs) play the role of request proxies and regional mobility anchors (LMAs) serve as content aggregation points.

C. Initializing a Default Distribution System

To illustrate the PANINI routing system in detail, let us consider an initial network prior to any signaling. This system consists of interconnected routers and a collection of NACs. NACs have prefixes assigned and routers preconfigured their FIBs autonomously to `exclude /* *` (cf., Table I). The initialization of the distribution system then proceeds in three phases.

1) Setting-up Defaults: Once configured, the NACs will start to announce their prefix availability and thereby construct shortest path trees (SPTs) per prefix. Efficient protocol mechanisms for that task are well known such as BIDIR PIM [26] in the Internet, or RPL [23] in the IoT. These SPTs are defined by the corresponding default entries in regular FIBs (cf., Table I). It is noteworthy that defaults enable any router on path to distinguish upstream and downstream messages prefix-wise.

2) Registering Content Names: On completion of Phase 1), content is only available at or via the NACs (e.g., from external peering). In order to publish content, providers need to register their content names with the appropriate NAC. For this providers issue a Name Advertisement Message (NAM) that is forwarded along the default route to its NAC. At each hop, intermediate routers see such advertisement and must update their FIBs to maintain consistency. For example, an 'exclude all' interface must change state to include the newly seen name or prefix. Alternatively, forwarding prefixes that have been already configured may need extension to include the new advertisement. Beyond consistency demands, an intermediate router is free to decide about the level of precision it includes in its Forwarding Information Base. To ensure message redundancy, resilience, and to facilitate adaptive FIB management, NAMs need periodic retransmission as is common in related Internet protocols.

3) Processing Content Interest: In the third phase, all content is available for request as described in Section III-A. Interests arriving at a router face are placed in the Pending Interest Table (PIT) like in regular NDN/CCN, and are forwarded under aggregation according to the FIB. However, as FIBs need not be complete, Interest forwarding needs to adapt in the following way.

Any Interest traveling up-tree will be forwarded along default routes, unlike a *specific* FIB entry (i.e., 'include' as in Table I) refers downwards. It should be noted here that longest prefix match cannot be applied to name-based routing without globally coordinated aggregation. At the NAC, a complete FIB will guide the Interest down to its dedicated subtree. Arriving on the downward path, any intermediate router will search its FIB for a specific route. If present, the Interest will travel in regular unicast mode. In case of a FIB miss, the router will select all down-tree interfaces without a matching exclude entry for broadcasting the Interest. We will show in Section IV that the expected broadcast fanout is small, and—by the recursive nature of SPTs—independent of network size. Note that loops are strictly prevented as Interests travel up-tree only once and monotonically downwards thereafter.

Additionally, Interest arrival is a measure of content popularity and used to adapt the FIB population at intermediate routers. We will discuss such adaptive FIB management in the subsequent section.

D. Adaptive FIB Management

A major objective of PANINI is to effectively limit FIB sizes. This is at first achieved by enhanced aggregation and default routes, but strict limits require additional measures. We now address how local routers can independently limit their FIBs in an optimized fashion.

A PANINI router can impose strict limits on its FIB, the minimum being a single default `/*`. While admitting incomplete forwarding information by flooding, it is the idea to keep broadcasts unlikely by populating the FIB according to content popularity, which is highly skewed (see Section II). PANINI does not dictate a common policy for managing FIBs, but rather leaves this to individual capacities and configurations of nodes. At the same time, any on-path router can measure name popularity through Interest processing and maximize the utility of its table entries.

We favour two strategies for an adaptive FIB management, leaving the field open, though, for further strategic improvements. A minimal approach—also feasible in the IoT—is to fix a table size and replace by least frequent use. In detail, FIB entries keep a statistic counter that is incremented for every Interest match. When a new name advertisement arrives, it replaces the least popular FIB entry, leaving the total size unchanged. Additional thresholds in frequency and time can increase convergence of this simple scheme.

A relaxed scheme based on soft states may be preferable at moderately constrained FIB sizes. Every advertised name will at first written to the FIB with a timestamp attached. A FIB entry then will expire after a timeout period, unless an Interest refreshes its timestamp. In this way, FIB tables will adjust to content variety and request frequency, possibly fluctuating heavily in size. The actual FIB properties may be fine-tuned by adjusting the timeout period or imposing additional thresholds.

IV. EVALUATION

In this section, we evaluate PANINI with a special focus on routing costs and overheads. We will approach the subject from two sides, theoretically by analysing the structural properties of the routing trees, and experimentally by emulating virtual PANINI networks in our lab. While theory grants rigorous insights into intrinsic characteristics of the system and to scale its size, experiments practically reveal net effects from the different constellations of the huge state space. Wherever possible, we compare results.

A. Theoretical Modeling

The PANINI routing scheme is built on prefix-specific shortest path trees (SPTs) that are rooted at the corresponding NACs. Without loss of generality it suffices to analyse the properties of a single tree, as no further assumptions are made w.r.t. individual prefixes.

1) Flooding Fanout: A router on the shortest path may experience a FIB miss in PANINI and needs to broadcast an Interest. In the absence of `exclude` entries, the flooding overhead increases with the fanout. Fanout resp. degree distributions are known for URTs [22]. Consider a URT of N nodes, then the expected number of nodes with degree k can be approximated by

$$E\left[V_N(k)\right] = \frac{N}{2^{k+1}} + \mathcal{O}((log\ N)^{k+1}). \tag{1}$$

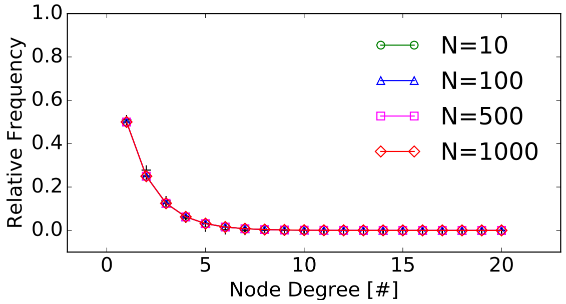


Figure 3. Node degree distribution for URTs of different sizes

Figure 3 shows the normalized degree distribution for different network sizes which coincide due to the recursive nature of the trees. In particular, fanouts (= degree - 1 uplink) are uniform and likely range between one and five. This confirms a limited ramification of URTs that are uniformly wide but short, and a broadcast at a single node will have limited multiplicity.

2) Flooding in Subbranches: When a PANINI node floods downwards after a FIB miss, there are two options of Interest propagation at each receiver. Either it holds a matching FIB entry, or it continues flooding. By the latter, flooding may extend over complete subbranches, which again share the URT property at reduced size. In this section, we calculate the worst, best, and average number of nodes that receive a flooded Interest message.

In the worst case, a direct child of the NAC holds no forwarding state for a specific Interest message and initiates flooding. In the absence of any further forwarding information, the nodes which receive the broadcast message are all in that branch. The expected size of such a branch can be calculated from the number of nodes and the expected root degree as follows.

Let U_N denote the random variable that maps from an URT with N nodes to the degree of the root vertex. Then the expected root degree reads [27]

$$E\left[U_N\right] = \sum_{j=1}^{N-1} \frac{1}{j} = H_{N-1} \tag{2}$$

that is the $(N-1)^{th}$ harmonic number H_{N-1}. Since the distribution of branch sizes is uniform in an URT, we obtain the expected branch size from dividing the remaining $N-1$ by (2).

A Uniform Recursive Tree of N nodes has an average depth of $\log(N)$, which is the optimal number of downtree Interest messages. For the average scenario, we consider the unicast hops and a random FIB on path empty with its corresponding subtree flooded (see below). We visualize the outcome in Figure 4. As can be seen from the graph, the average number of Interest messages needed for locating content follows closely the logarithmic behavior of the best case. In contrast, the worst case scenario grows almost linearly ($\approx N/\log N$).

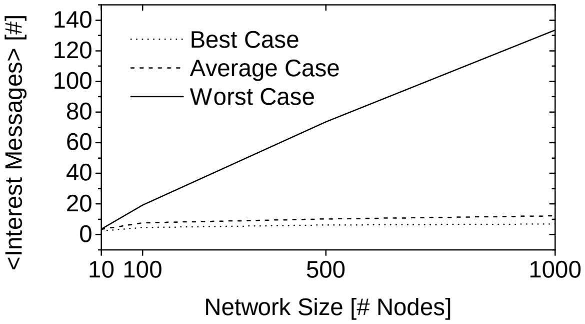


Figure 4. Average number of Interest messages for different scenarios and network sizes

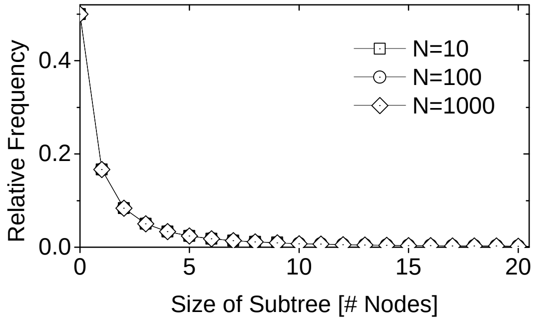


Figure 5. Distributions of branch sizes in a routing tree of N nodes

We now calculate the size distribution of a randomly selected subtree following the Polya urn model. We consider N nodes that are numbered in the order of attachment. Let $D_N(k)$ denote the number of descendants of node $k > 1$, then the distribution can be derived from the Polya-Eggenberger distribution [28], [29]

$$\mathcal{P}\left(D_N(k) = j\right) = \frac{(k-1) \cdot (N-k-j+1)^{\overline{j}}}{(N-j-1) \cdot (N-1)^{\overline{j}}}, \tag{3}$$

with $(n)^{\overline{j}}$ the j-th rising factorial power of n.
Summing over all nodes k with equal probability $1/N$ yields the distribution D_N of nodes in a subtree rooted at an arbitrarily chosen node.

Figure 5 visualizes these analytical distributions for different numbers of nodes. Strikingly, the branch sizes are largely independent of the overall network size, which is due to the recursive nature of the URTs. Node numbers from these exponentially decaying distributions are rather small: more than 7 nodes appear with probability 0.01. This is again due to a uniformly wide fanout—trees are rather wide and short.

3) Resilience and Robustness: To provide robust data distribution a network needs to adjust to changes and failures. PANINI operates on the basis of shortest path trees and needs to cope with nodes that disappear from the SPT and possibly reappear somewhere else on the tree at a later time. Therefore, the FIBs need to be updated, e.g., by periodically repeating publishing messages. With

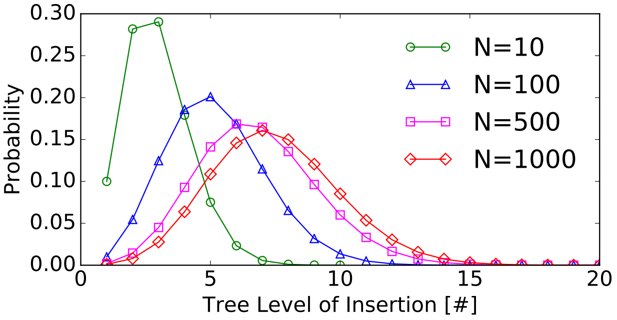


Figure 6. PDF of the insertion depth of a failing node

the insertion depth of a node in an URT, we can estimate how many link changes are required and thereby how large a damage and the effort to update the FIB at the NAC are.

To obtain the insertion depth of a node, we use the profile $E\left[V_{d,N}\right]$ of a URTs [22], which describes the expected number of nodes at level d. Let random variable I_N map to the insertion depth d of a node, then

$$P(I_N = d) = \frac{E\left[V_{d-1,N-1}\right]}{N-1} \qquad (4)$$

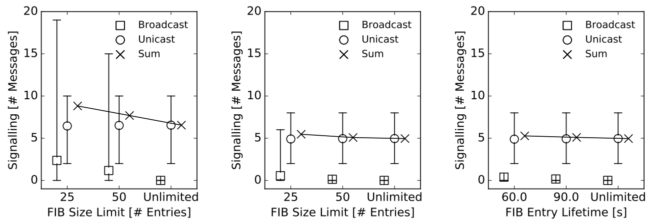
Figure 6 depicts the probability density functions of I_N for different network sizes. The graph shows that the insertion depth of a node only slowly increases with growing size. The reason is again that the shapes of URTs tend to grow relatively wide but not very deep. Furthermore, in URTs about half of the nodes will be situated at level $\ln(n)$. That means that we expect a new publish message to travel $\ln(n)$ hops until the NAC has the right information in which subtree the node rejoined.

B. Experimental Emulation

We now proceed to our experimental evaluations that are performed using a realistic emulation environment. While the theoretical analysis remained limited to structural properties of the routing system, we experimentally consider real-world topologies, realistic name popularities, and adaptive FIB management. Caching was not considered in this work, since arbitrary cache effects would blur the outcome of routing and FIB management that we want to reveal. However, adding caching to the system will only improve the overall performance.

Evaluations again concentrate on the routing costs and overheads produced by PANINI. In addition to control messages and flooding costs, we quantify the path stretch for the Rocketfuel topologies, which also serves as an indicator for overheads in forwarding delay.

1) Experimental Environment: For experimentation, emulated networks were set up on a 64-core host machine based on virtual nodes and Mininet [30]. Topologies were created from Rocketfuel data [31] and from artificially generated URTs.



(a) URT (b) AS 1239 (sized) (c) AS 1239 (timed)

Figure 7. Overall distribution of message types for different FIB sizes resp. FIB entry life times, showing average, 95% and 5% percentile

Every virtual node ran an NDN-Forwarder based on a modified version of NDN (0.4.0-beta2)[4]. PANINI modifications were implemented in the NDN strategy layer which allows for specific processing of each individual packet. We implemented a fixed size FIB operating a LFU replacement policy and a life-time management for FIB entries. Using this setting, we were able to reliably run networks of several hundred core routers without loosing packets or exhausting resources otherwise.

All individual experiments were performed according to the following scheme. We fixed a topology and FIB size, and placed the NAC at its center (on the node of highest betweeness). For each content name from a set of 10.000, we placed providers on random but fixed routers in the topology. Consumers were emulated as child nodes of the core routers. Each consumer was placed uniformly random and requested content from our name set according to a Zipf distribution. One million individual message paths were iterated, monitored, and recorded for evaluation.

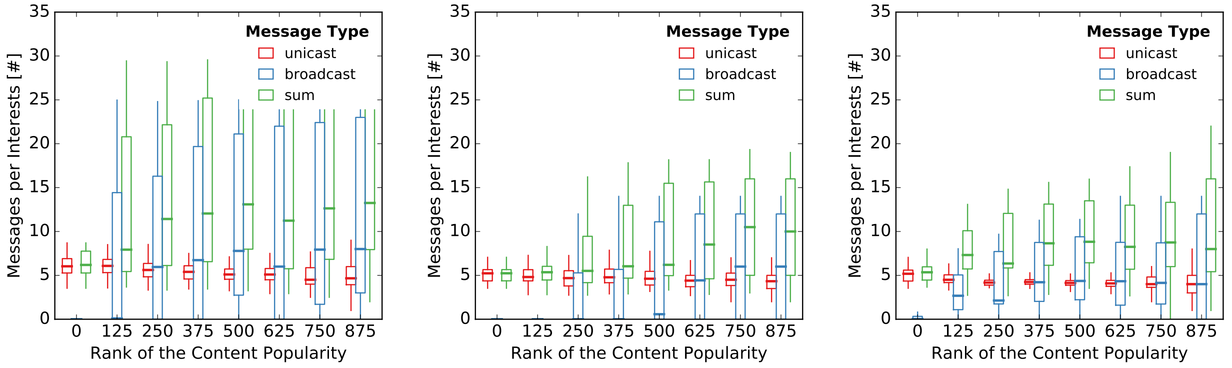
C. Experimental Results

1) Expected Message Distribution: Our first glance at the system addresses the overall messaging behaviour. We are interested in the average number of unicasts and broadcasts per content request for different topologies and FIB sizes, as well as FIB entry life times. The results in Fig. 7 surprise with a remarkably low broadcast appearance for both, the artificial URT (size 100 routers) and the AS1239 Rocketfuel topology (size 315 routers). Only for small FIB sizes in the URT, broadcast multiplicities fluctuate by an order of magnitude. With increasing FIB sizes, but in particular for the temporal FIB entry management, flooding reduces to about a single broadcast per request. Note that the routing refresh rate equals 200s and the average path lengths in both topologies is close to six.

For a more differentiated view, we correlated the message type distribution with content ranks (see Fig. 8). Unsurprisingly, broadcast multiplicities and fluctuations largely increase with decreasing popularity. In detail, content requests above rank 100 caused visibly fluctuating broadcasts. However, given the heavily skewed content distribution

---

[4] http://named-data.net/



(a) URT (b) AS 1239, FIB size limit 50 (c) AS 1239, FIB Entry Lifetime 60s

Figure 8. Measurements of (typed) message frequencies for selected popularity ranks and different scenarios

known from Fig. 1, it becomes evident that these flooding events carry limited weight and contribute little to the overall average results in Fig. 7.

These numbers strongly support the initial PANINI assumption that low-ranked content names in FIBs are of little use.

2) Flooding in Subtrees: The following analysis explores the empirical shapes of flooded subtrees, which had already been discussed in the theory section. For the same experiments, we identify (a) all coherent subtrees and (b) the accumulated size of flooded regions that are composites of a larger tree with subtrees connected via unicast links. The latter are results from alternating FIB misses and hits along paths.

Fig. 9 visualizes the different distributions of subtree sizes. Results for single broadcast trees only, are in excellent agreement with theory and very small. Almost 90 % of flooded subbranches subsume less than 10 nodes. This again reflects the limiting characteristic of the recursive structures. On the contrary, composite trees tend to be much larger with about 50 % exceeding 25 network nodes. This is an indication of fluctuating decision at neighboring routers that cannot converge on treating certain names. Even though these events occur rarely and carry little weight, we expect to improve this behavior with name aggregation at FIBs close to the NAC. Name aggregation has not been implemented yet, but shall assign an increasing weight of names at up-tree routers.

3) Path Stretch: In most cases, PANINI transmits Interests via a NACK to the producer (in the absence of caches) and thus may artificially extend paths. To quantify this effect, we evaluate the distributions of path stretches for all Rocketfuel topologies. The results are displayed in Fig. 10.

Strikingly, 50 % of all paths experience no stretch at all except for the slightly outlying AS 1239 topology. Larger stretches exceeding two are very rare—mainly in less than 10 %. These results are tightly connected with the placement of NACs. Being at the center of the network, many shortest paths traverse through the NAC and experience no stretch. Caching at NAC will improve the results even further.

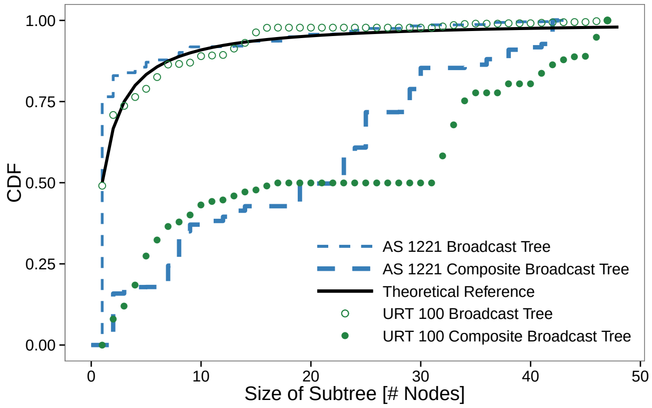


Figure 9. Distribution of subtree sizes for single and composite broadcast trees at FIB size 50—comparison of measurements from two topologies with theory

V. CONCLUSIONS AND OUTLOOK

Name based routing and forwarding in Information Centric Networking offer interesting potentials, but continue to raise significant challenges. In this work, we proposed a way to limit routing table sizes and to benefit from name aggregation within topological constraints. We introduced and thoroughly analysed PANINI, which may lead a new way to simplified content networking. Experimental as well as a theoretical evaluations revealed promising results in several dimensions.

In summary, we could show that PANINI routing is a self-optimizing hybrid approach that mitigates between FIB sizes and Interest flooding while locating content. Evidence was presented that rigorously small, incomplete routing tables can be compensated by a negligible quantity

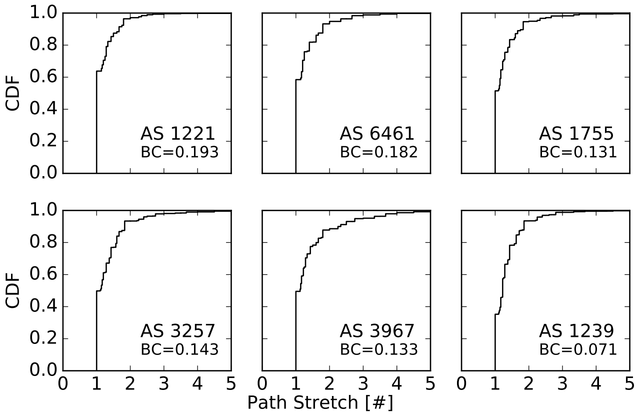


Figure 10. Distribution of path stretchs for PANINI Routing in different Rocketfuel topologies, BC denotes the Betweenness Centrality of the NAC

of broadcasts. Our future work will concentrate on to elaborate and quantitatively evaluate the aggregation potentials in distributed name-based routing. For the default free zone, this will raise the particular challenge of a scalable name synchronisation at interdomain peering. Corresponding routing strategies need to be found. It is our intend to show the feasibility of PANINI even for large-scale inter-provider settings.

REFERENCES

[1] D. Kutscher, S. Eum, K. Pentikousis, I. Psaras, D. Corujo, D. Saucez, T. Schmidt, and M. Waehlisch, "ICN Research Challenges," IETF, Internet-Draft – work in progress 06, March 2016.

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

[3] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A Survey of Information-Centric Networking," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, July 2012.

[4] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A Survey of Information-Centric Networking Research," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.

[5] T. C. Schmidt, S. Wölke, N. Berg, and M. Wählisch, "Partial Adaptive Name Information in ICN: PANINI Routing Limits FIB Table Sizes," in *2nd ACM Conference on Information-Centric Networking (ICN 2015), Poster Session*. New York: ACM, Oct. 2015, pp. 193–194.

[6] Y. Chung, "Distributed Denial of Service is a Scalability Problem," *SIGCOMM CCR.*, vol. 42, no. 1, pp. 69–71, Jan. 2012.

[7] M. Wählisch, T. C. Schmidt, and M. Vahlenkamp, "Backscatter from the Data Plane – Threats to Stability and Security in Information-Centric Network Infrastructure," *Computer Networks*, vol. 57, no. 16, pp. 3192–3206, Nov. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2013.07.009

[8] M. D'Ambrosio, C. Dannewitz, H. Karl, and V. Vercellone, "MDHT: A Hierarchical Name Resolution Service for Information-centric Networks," in *Proc. of the ACM SIGCOMM WS on ICN*. New York, NY, USA: ACM, 2011, pp. 7–12.

[9] K. V. Katsaros, X. Vasilakos, T. Okwii, G. Xylomenos, G. Pavlou, and G. C. Polyzos, "On the Inter-domain Scalability of Route-by-Name Information-Centric Network Architectures," in *Proc. of IFIP Networking*, 2015.

[10] T. Song, H. Yuan, P. Crowley, and B. Zhang, "Scalable name-based packet forwarding: From millions to billions," in *Proc. of ACM ICN*. New York, NY, USA: ACM, 2015, pp. 19–28.

[11] J. J. Garcia-Luna-Aceves, "A More Scalable Approach to Content Centric Networking," in *ICCCN*. Piscataway, NJ, USA: IEEE, 2015.

[12] A. Afanasyev, C. Yi, L. Wang, B. Zhang, and L. Zhang, "SNAMP: Secure Namespace Mapping to Scale NDN Forwarding," in *Proc. of IEEE Global Internet Symposium*. Piscataway, NJ, USA: IEEE, 2015, pp. 281–286.

[13] L. Wang, S. Bayhan, J. Ott, J. Kangasharju, A. Sathiaseelan, and J. Crowcroft, "Pro-Diluvian: Understanding Scoped-Flooding for Content Discovery in Information-Centric Networking," in *Proceedings of the 2nd International Conference on Information-Centric Networking*. ACM, 2015, pp. 9–18.

[14] R. Kleinberg, "Geographic Routing Using Hyperbolic Space," in *INFOCOM*. Piscataway, NJ, USA: IEEE Press, 2007, pp. 1902–1909.

[15] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguñá, "Hyperbolic Geometry of Complex Networks," *Physical Review E*, vol. 82, no. 036106, Oct 2010.

[16] D. Papadimitriou, D. Colle, P. Audenaert, and P. Demeester, "Geometric Information Routing," in *Proc. of IEEE ANTS*. Piscataway, NJ, USA: IEEE, 2013.

[17] P. Barford, A. Bestavros, A. Bradley, and M. Crovella, "Changes in Web Client Access Patterns: Characteristics and Caching Implications," *World Wide Web*, vol. 2, pp. 15–28, 1999, special Issue on Characterization and Performance Evaluation.

[18] M. Halvey, M. T. Keane, and B. Smyth, "Mobile Web Surfing is the Same as Web Surfing," *Commun. ACM*, vol. 49, no. 3, pp. 76–81, 2006.

[19] G. K. Zipf, "Relative Frequency as a Determinant of Phonetic Change," *Harvard Studies in Classical Philology*, vol. 40, pp. 1–95, 1929.

[20] P. Van Mieghem, G. Hooghiemstra, and R. van der Hofstad, "A Scaling Law for the Hopcount in Internet," Delft University of Technology, Tech. Rep., 2000.

[21] P. Van Mieghem, *Performance Analysis of Communications Networks and Systems*. Cambridge, New York: Cambridge University Press, 2006.

[22] M. Drmota, *Random Trees: An Interplay between Combinatorics and Probability*. Springer Science & Business Media, 2010.

[23] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," IETF, RFC 6550, March 2012.

[24] E. Baccelli, C. Mehlis, O. Hahm, T. C. Schmidt, and M. Wählisch, "Information Centric Networking in the IoT: Experiments with NDN in the Wild," in *Proc. of 1st ACM Conf. on Information-Centric Networking (ICN-2014)*. New York: ACM, September 2014, pp. 77–86. [Online]. Available: http://dx.doi.org/10.1145/2660129.2660144

[25] T. C. Schmidt, M. Wählisch, and S. Krishnan, "Base Deployment for Multicast Listener Support in Proxy Mobile IPv6 (PMIPv6) Domains," IETF, RFC 6224, April 2011. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6224.txt

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

[27] Q. Feng, C. Su, and Z. Hu, "Branching Structure of Uniform Recursive Trees," *Science in China Series A: Mathematics*, vol. 48, no. 6, pp. 769–784, 2005.

[28] N. Johnson and S. Kotz, "Urn Models and Their Application: An Approach to Modern Discrete Probability Theory," 1977.

[29] A. Panholzer and H. Prodinger, "Level of Nodes in Increasing Trees Revisited," *Random Structures & Algorithms*, vol. 31, no. 2, pp. 203–226, 2007.

[30] N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown, "Reproducible Network Experiments Using Container-based Emulation," in *Proc. of CoNEXT '12*. New York, NY, USA: ACM, Dec. 2012, pp. 253–264.

[31] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP Topologies With Rocketfuel," *IEEE/ACM Trans. Netw.*, vol. 12, no. 1, pp. 2–16, 2004.