Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Measuring and Evaluating Efficiency and Scalability of the PANINI Approach to Name-Based Routing

Sebastian Wölke

Writing Report, Project II

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Contents

1 Introduction

Information Centric Networking (ICN) introduces a promising network communication approach for the Internet of the future. The idea of ICN is to request content by unique names rather than querying hosts for stored content as in today's Internet. This new paradigm is providing several inherent features like in-network caching, group communication, and mobility, which requires much more effort in current IP based networks. On the other hand, this research field also openes new challenges [1]. The challenge we are focusing on is the design of a scalable name-based routing approach. Naming of content leads to several magnitudes of more names than IP addresses, which have to fit in Forwarding Information Bases (FIB) of routers. Several proposals exists [2], [3], [4] to compress and aggregate FIBs but caused by the sheer scalability demands further research is needed.

The approach Partial Adaptive Name Information in ICN (PANINI) [5], [6] faces the challenge on multiple levels. PANINI introduces aggregation points to (1) divide the namespace and its routing cost among multiple routers, and (2) aggregate names in a hierarchical network. In addition, PANINI (3) allows to limit FIB-sizes of routers, and (4) can still be used orthogonal to FIB compression techniques. An aggregation point is a name-based router, which is responsible for a set of content names and manages registrations of content suppliers. Content requests are forwarded to its aggregation points, which are located at the root of a name-specific shortest path tree (SPT). Such a SPT is previously constructed by a topology building mechanism initiated by aggregation points. Then, the requests are forwarded down the tree to their respective content suppliers. Shortcuts between requester and suppliers are possible. PANINI can limit FIB-sizes of routers by smartly dropping entries and flooding respective requests. The idea is to exploit the diversity of the content popularity in the Internet and drop seldom used entries to keep the flooding rate low.

In this work, we measured and evaluated efficiency and scalability of PANINI. We implemented PANINI and integrated it in the Name Data Networking (NDN) project [7]. Then, we tested and measured it in a real-world scenario. We used Mininet [8] to simulate different topologies and equipped the router with our implementation. In our investigation we focused on metrics flooding rate, path stretch caused by the constructed shortest path tree, and FIB-size usage. We can show that the overhead based on control traffic and path stretch is moderate and the advantage of limiting the FIB-size outweighs flooding cost.

The remainder of this paper is organized as follow. In Chapter 2 we recapitulate CCN the most promising ICN architecture, we discuss different name-based routing approaches, and present the PANINI approach. Our implementation and its integration in NDN is described in Chapter 3. In Chapter 4 we illustrate our measurement and test setup. Our measurement results are summarized and discussed in Chapter 5. Finally, we give a conclusion and an outlook.

2 Background and Related Work

Information Centric Networking (ICN) is a Future Internet concept, which we describe in detail in Section 2.1, with a focus on name-based routing. We show differences to IP-based routing in Section 2.2, point out a problem related to prefix aggregation in Section 2.3 and give an overview of FIB-size reduction concepts in Section 2.4. Finally, we focus on our approach PANINI. In Section 2.5, we show the diversity of the content popularity, which we exploit in our name-based routing approach PANINI, described in section 2.6.

2.1 Information Centric Networking

Information Centric Networking (ICN) is a novel communication paradigm and follows the idea of labeling content with a unique name for identification. Names are location independent and favor concepts like in-network caching, group communication, and mobility. In contrast, the current Internet model is host-centric and designed for a pair-wise communication. In a pair-wise communication transmitted packets are only valid together with the context of the sender or the receiver, which makes it difficult by design to re-use transmitted data or adapt state changes of hosts. Early proposals for ICN based networks are TRIAD [9] and DONA [10]. Nevertheless, the majority of current work follows the idea and concept of Content Centric Networking (CCN) [11].

CCN relies on a hierarchical namespace and the message types *Interest* and *Data Packet*. The Interest message is sent by a content requester (consumer) and the Data Packet is sent by a content provider (producer) or a cache in response of an Interest. A forwarding engine acts as a network stack between applications and networks, it routes Interests to sources, and it forwards respective Data Packets along the reverse path to the requester. It provides faces, a generalization of a network interface, for a transparent communication channel between applications and networks of arbitrary technology. The forwarding engine consist of three basic components, a *Pending Interest Table* (PIT), a *Forwarding Information Base* (FIB), and a *Content Store* (CS). The PIT stores Interests with their incoming face to enable reverse path forwarding (RPF) for matching Data Packets. PIT entries are deleted after usage or time out if no matching Data Packets were received. The FIB is similar to the FIB of an IP-Router, it forwards Interests to potential sources and is filled manually or automatically for example by a routing protocol. Finally, the CS caches received Data Packets. These CCN entities are the basis for the forwarding engine and are processed in the order CS, PIT, and FIB in an forwarding procedure of an Interest. At first, an incoming Interest is matched with CS for a respective Data Packet. If a Data Packet is available, it is send out to the incoming face of the Interest. Otherwise, the forwarding engine moves to the next step. The PIT is checked for a similar stored Interests. If an entry is already available, the FIB entry is updated by adding the incoming face to the existing entry. Otherwise, a new entry is added and the Interest is forward based on the FIB to one or more faces. In addition, a *Strategy Layer* is intended to handle different Interest forwarding strategies. For example a Multicast Strategy could forward an Interest to multiple faces to increase the chance of finding a Content Store with the answer near-by. Another strategy could focus on high throughput or short delay.

Available implementations for CCN are CCNx [12], CCN-lite [13], and NDN [14]. CCNx provides an ICN framework and is working to standardize different components of its implementation [15] at the IETF. CCN-lite is a lightweight and interoperable implementation to CCNx and NDN (Named Data Forwarding). In this work, we focus on NDN which provides several implementations like a Forwarding Daemon (NFD), a Dataset Synchronisation Library (ChronoSync) [16], and the routing protocol NLSR [17]. NLSR is an intra-domain routing protocol derived from the Link-state routing protocol OSPF [18]. It supports NDN features like multipath forwarding and NDN build-in security.

2.2 Control and Data Plane

A network architecture can be divided into control and data plane. The control plane provides the knowledge of where to forward packets to reach their destinations. The data plane or also called forwarding plane uses this information to handle packets accordingly. In the current Internet design, the control plane of a router is stateful and creates its knowledge by provider configuration and dynamic routing protocols. The data plane of a router is decoupled from its control plane, stateless, and forwards or drops data packets based on the gathered information of the control plane.

In contrast, ICN shifts the Internet to data awareness. The control plane still provides the knowledge of where to forward packets (Interests). However, the data plane is stateful and coupled with the control plane [19], [20]. In detail, content can be registered by any producer, which causes an update of the control plane. This contrasts with the current Internet design, where routing and DNS states are not changed on adding new content to a web page or on caching data. The ICN data plane holds data-driven states as for each data packet reverse forwarding information have to be stored [21].

2.3 Longest Prefix Matching

Longest Prefix Match (LPM) is an algorithm used by IP routers [22]. It is used together with prefix aggregation to compress FIBs. The destination address of an incoming data packet is matched with the most specific entry in the routing table to determine the outgoing interface. This is possible as IP addresses are enumerable and therefore, prefix aggregation is locally decidable. The LPM algorithm cannot be easily mapped to name-based routing, as names are not enumerable. An incorrect prefix aggregation is referred to *suffix hole* [23] and is illustrated in Figure 2.1. The network consist of six router, R1 to R4 are located in the default-free Zone (DFZ) and R5 and R6 are located outside of the DFZ. For simplification it is assumed that each router in the DFZ performs an exact matching algorithm on the FIB and entries are not aggregated. Consumer 1 and consumer 2 are interested in content of producer 2 (*/google/maps*). Interests of consumer 1 are routed correctly, as it is directly connected to R1. On the other hand, consumer 2 is physically connected to R5 as well as producer 2, which provides content for prefix */google*. R5 is not part of the DFZ, but physically connected to it. R5 has two FIB entries, a default path to DFZ (*/**) and an entry to producer 1 (*/google*). In this case, Interests of consumer 1 are routed mistakenly to producer 1 instead to producer 2, as the most specific entry chosen by LPM is */google*. On designing name-based routing protocols it is imported to note that prefix aggregation, as commonly used in IP networks can be a pitfall. Even if protocols have solved or avoid this pitfall, on linking two protocols (for example an intra domain protocol and an inter domain protocol) the problem can reoccur.

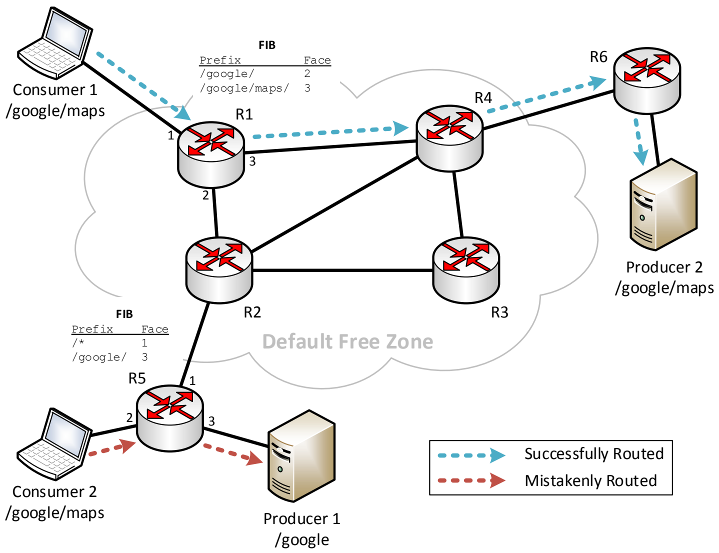


Figure 2.1: The Interest of consumer 2 (*/google/maps*) is routed by R5 mistakenly to producer 1 (*/google*) instead to producer 2 (*/google/maps*).

2.4 FIB Aggregation and Compression

FIB-size of ICN routers "requires several orders of magnitudes more storage than the corresponding IPv4 tables" [24]. In detail, current BGP routers have to handle $6 * 10^5$ entries [25] , which cover over 10^9 IPv4 hosts [26] by aggregation. A transition to name-based routing with similar techniques would blow current FIBs. The DNS alone has over $3 * 10^8$ unique names [26], where prefix aggregation is not possible. Furthermore, in the year 2008, *Google* announced that its Index stores 10^{12} web addresses [27] and the paper A Hierarchical Name Resolution Service (MDHT) [28] even expects 10^{15} names. Therefore, additional aggregation and compression techniques are needed to reduce FIB-sizes in ICN.

The paper *Scalable Name-Based Packet Forwarding* [29] proposes a data structure to compress the FIB and speed up the lookup. The data structure is called *Dual Binary Patricia* (DuBP) and consists of two modified binary *Patricia Tries* with an additional compression level. The

Tokenized Binary Patricia Trie (tBP) handles flat names and the *Speculative Binary Patricia Trie* (sBP) is responsible for the prefix subset. A Patricia Trie [30] is basically a compressed search tree, where each child node is merged with its parent node, when the parent has exactly one branch. The additional compression level is added by basically removing all these children but the first, instead of merging them. It is still possible to lookup names by a fuzzy search, but the Trie losses the ability to recover origin entries and to recognize if a name is not available. To compensate this disadvantage, DuBP is proposed together with a forwarding behavior called *Speculative Forwarding* in a two-level network architecture. The first level is a default-free zone (DFZ) running the Speculative Forwarding behavior managed by routers, which are using DuBP as FIB. The second level consist of local networks with conventional routers connected to the DFZ. Speculative Forwarding performs Interest forwarding properly, if an FIB entry for the Interest exist. While Interests without a matching entry are forwarded randomly through the DFZ and are only dropped if a loop is detected or if they reach an edge router. The strength of this proposal is a well compressed FIB, which is moreover independent of the length of names. On the other hand, Speculative Forwarding can be easily exploited for DoS attacks and router in the DFZ and router in local networks are facing the prefix aggregation problem in named-based routing (chapter 2.3).

The paper *CONCERT: Constructing optimal name-based routing tables* [31] reduces the number of stored prefixes in a Patricia Trie FIBs to a minimal number. In a Patricia Trie prefix names are saved in edges and face IDs are saved in nodes. This means, when an Interest has a longest prefix match, the matching algorithm ends at a specific edge of the Patricia Trie and the Interest is forwarded to the face ID saved in the attached note of this edge. It is very common that a parent node of the attached node has multiple children saving the same face ID. CONCERT proposes to minimize this redundancy. The optimal FIB is basically constructed by traversing every parent and for each parent removing all children with the most frequent face ID and adding a default entry for this face ID, instead. If an Interest cannot be matched, as it is removed by the optimization and the default entry is used. The strength of CONCERT is that it has a good compression rate and can be used orthogonally to other compression techniques like a Bloom filter approach [32] to shrink a FIB to fit in the fast RAM of a router to increase the lookup speed. On the other hand, this algorithm still requires a reference of the full FIB to stay in an optimal state after an incremental update operation.

The paper *VDR: Virtual Domain-based Routing Scheme* [33] proposes to distribute the global FIB to virtual domains and sub-domains to reduce FIB-sizes on routers. A virtual domain is responsible for a subset of routing names. Each virtual domain is bound to a hash value and processes all names which are hashed to this value. For example, the Interest *facebook/simpson/images* is received on an edge router. The edge router hashes the first component to the domain name */D1* and forwards the Interest to this domain. Afterwards, the second component is hashed to the value *D11* and the Interest is forwarded from domain */D1* to domain */D1/D11* and so on until a sub-domain knows where to find the producer or a cache with the respective content. Virtual domains and sub-domains of all levels are bound to physical routers, whereby a router can be part of no, one, or several domains. In the current state of this routing approach it is assumed that each router knows how to route to any other router. This approach has the trade-off between path stretch and FIB-size, as for a small FIB-size the number of virtual domains a router is part of must be low. But the probability of a detour for Interests increases the less often a domain with its routing information are shared among physical routers. The strength of this proposal is a distributed global FIB, which reduces the FIB-size of an individual FIB. On the other hand, this work is in an early state and leaves some questions unanswered, for example how is the trade-off between path stretch and FIB-size or how are routers assigned to a virtual domain.

2.5 Name Popularity

A name popularity model shapes the frequency of requested content in a measurement scenario. While the content popularity distribution has a big influence on the efficiency of PANINI we are looking for a realistic model. An often used model for this purpose is the power law distribution Zipf's law [34]. It was confirmed in the past for Internet objects like web caching [35] and website usage [36] but refute for user generated content like Youtube videos [37] or peer-to-peer data [38]. Instead of simply use Zipf's law we wanted to verify if it still applies for our purpose. We verified it empirically by analyzing the website usage of the Internet as it has the largest coverage of all metrics.

For the verification we gathered information provided by three different web analytic services. **Alexa**[3] an *amazon.com* company provides a free and global top one million website list which we used as reference list for further investigations. Furthermore Alexa provides an API to get detailed information like relative pageviews over the time ranges months, weeks, and

[3] http://www.alexa.com/

days for individual websites. We additionally crawled the website of **Quantcast**[1] which offers a free list of websites with their unique monthly visits of the United States. Finally we crawled the **Informer**[2] website based on our Alexa reference list and gathered the information of the estimated absolut daily visitors and pageviews per website.

In Figure 2.2 we compared our dataset with Zipf's law. All curves follow a similar pattern to Zipf's law, even though all curves differ in their skewness, top, and the long tail. The curve of Alexa fits best, but drops on the long tail to zero as the API does not provide more accurate data for lower ranks. The varieties between Alexa, Quantcast, and Informer can be explained by the different metrics they provide. Quantcast provides a distribution of unique visitors where repeated pageviews of the same user are eliminated. This reduces the weight of the high ranked websites as their rank relies on the usage of many users which also use its service in a high rate (e.g. google.com). Cause of this eliminated factor the skewness is lowered which is represented in the log-normal scaled graph by a shift up the Y axis. Informer estimate its data based on third-party web analytic services and it is unclear why it differs from Alexa. However, in conclusion we were able to show that Zipf's law follows a sufficiently realistic pattern and can be used as a name popularity model in our measurement scenarios.

2.6 PANINI

PANINI (Partial Adaptive Name Information in ICN) is a name-based routing approach, which distributes the routing load and allows to limit FIB-sizes of routers. The routing load is distributed by introducing *Name Collectors* (NAC), which divide the namespace and its routing cost. The FIB-size can be limited on a router, by smartly drop FIB entries. Interests, which then get a FIB miss are simply flooded. This strategy still scales, as flooding areas are restricted and the diversity of the content popularity (see Section 2.5) is exploited by only dropping the least frequently used FIB-entries.

Multiple NACs can be placed in a network. Each NAC is responsible for a disjoint set of prefixes, which together cover the whole namespace. NACs act as aggregation points for its prefix set and are the only entities which require a complete FIB and can be seen as a default-free zone (DFZ) router. A NAC announces its position and advertises its prefix set. Then, each

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

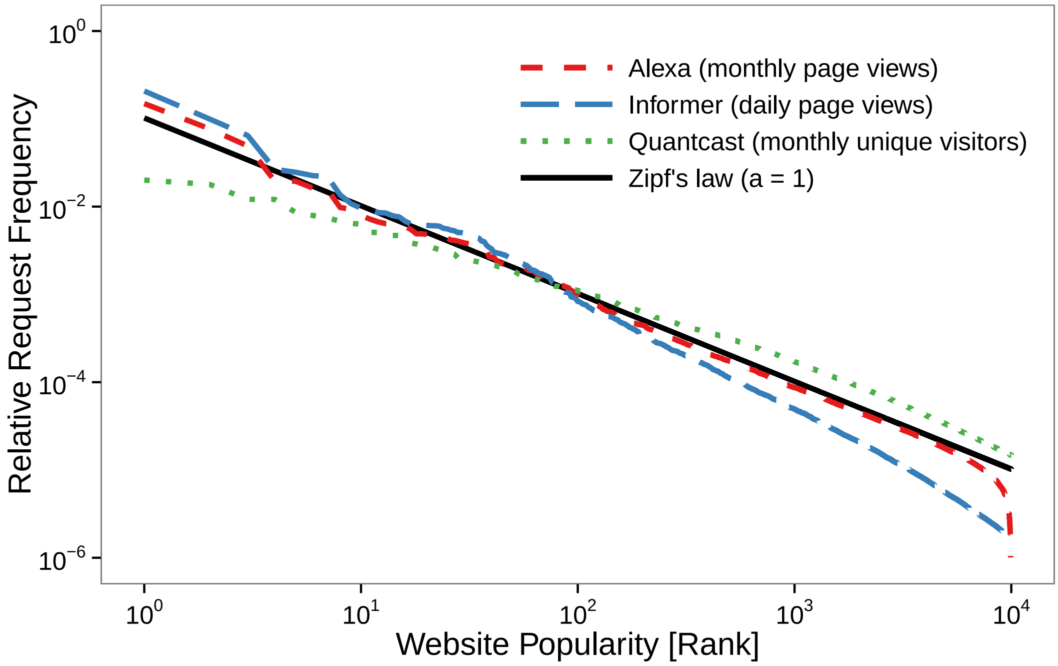


Figure 2.2: Zipf's law in compression to measured name popularity distribution from different sources.

router sets prefix-specific default paths to reach this NAC. The prefix-specific default path is the shortest path to the NAC and is constructing together with all other shortest paths of the same prefix a shortest path tree (SPT). Instead of advertising prefixes, a hash-routing approach [39] could be used to map content identifiers to NAC positions. This reduces the potential memory demand as FIB entries for prefix-specific default paths can be avoided.

Producers register their content at a NAC by sending a Name Advertisement Message (NAM). These NAMs are forwarded up the tree along the shortest path to the NAC. Each router along the path decides based on its available resources whether or not to save the path to the producer. All paths to producers are directed downwards, while prefix-specific default paths are directed up to the root of the SPT. Interests, send by consumers, are forwarded along the prefix-specific default path up to the NAC or up to a router with a matching forwarding entry added by a NAM. Then, Interests are forwarded down the tree to the producer. If a router has no forwarding information for an Interest, it is flooded to all Downstreams, instead.

Table 2.1: 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

PANINI also introduces improvements to the FIB, which reduces the FIB-size and minimize the flooding area. A PANINI FIB has the modes *default*, *include*, and *exclude* as shown in table 2.1. The include mode is filled by NAM messages and is used to forward Interests received on upstream side as well as on downstream side of the router. The exclude mode is intended to prune flooding of Interests in branches where no producer for a specific prefix exists. The default mode stores prefixes advertised by NACs and is used to forward Interests received at a downstream face as well as NAMs received anywhere. Furthermore, PANINI distinguish between two types of routing names, names with full prefix information (*/test/a/*) and names with one ore more aggregated prefixes (*/test/a/**). Entries with an asterisk can be used for LPM and entries without an asterisk have to be matched exactly.

2.7 Prefix Aggregation in Access Networks

Prefix aggregation can reduce FIB-sizes but it is not applicable in all situations. PANINI is designed to aggregate prefixes on upstream faces. Aggregation on downstream faces is only feasible when names are bounded to the topology as an intermediate PANINI router has incomplete information of its network. In detail, if names are bounded to the topology prefix aggregation is locally decidable (see section 2.3). An bounded name set can be enforced when names and their producers are controlled by a single authority.

In this section, we focus on a scenario where PANINI is deployed in an access network and discuss where prefix aggregation can be used. An access network, also called Eyeball network, provides Internet access to private users and companies. In most cases an bounded name set cannot be enforced and prevents the use of prefix aggregation on downstream faces.

Companies often host their own services and have to publish their content. Associated content names have no aggregation potential as published names of companies usually share no common prefix. In contrast, private users mostly consume content or publish content through third-party supplier like social networks or cloud services. In these cases, no content has to be published and therefore no additional FIB entries are required. Current Internet services that do not fit into these classes are based on technologies like peer-to-peer or WebRTC [40], [41].

Peer-to-peer content is shared directly without a third-party supplier. Whether peer-to-peer content has to be published depends on its ICN implementation and the functionality provided by the ICN network. A vanilla approach for a peer-to-peer network based on the ICN paradigm can be implemented as follows. A peer downloads a content chunk and afterwards published this chunk to make its own copy available for further peers. This approach leads to huge FIB-sizes as prefix aggregation cannot be used. Another solution which does not require registration of content but benefits from the in-network caching feature of ICN could rely on the Forwarding Label idea [42]. The Forwarding Label approach adds a locator name to the Interest and enable forwarders to forward Interests to a dedicated network or a host. In this case, host identifiers would be assigned and controlled by Eyeball ISPs and would allow prefix aggregation. In an ICN network where hosts can be addressed directly, a peer-to-peer network can operate similar to todays implementation without causing scaling problems and with improved content dissemination.

WebRTC enables browser-to-browser communication, is especially designed for realtime applications, and shares content without a third-party supplier. In an ICN environment, WebRTC can be integrated similarly to peer-to-peer applications and benefits from the in-network caching feature.

In summary, private users usually do not publish content and do not enlarge FIBs. This could change in the future as technologies like Internet of Things (IoT) are on the rise. The idea of IoT is to connect everyday things like temperature sensors and light bulbs to the Internet. This technology provides new content which may have to be published. IoT things of private users are most likely connected to a gateway which preprocesses sensor data. This gateway could publish extracted content or transmit the content to a cloud service. In the first case, additional FIB entries are required which cannot be aggregated as bounded topology names cannot be enforced. In the second case, FIBs of the access network remain unaffected.

3 PANINI Implementation

We integrated PANINI in the *Named Data Networking* (NDN) software [7] [14]. The NDN community provides a comprehensive open source toolbox containing a forwarding daemon, different client libraries, test and debugging tools, example programs and further more. We implemented the behavior of PANINI in the provided NDN forwarding daemon (NFD), and used the client library *NDN-cxx* for consumer and producer. Furthermore, we added different policies to limit the FIB-sizes. With this setup, we want to observe and analyze the behavior of PANINI in a real-world scenario.

3.1 NDN Forwarding Daemon

The NFD [43] is written in C++11 and implements an ICN forwarder and is complaint to the NDN protocol [44], which specifies the behavior and binary structure of different NDN entities like Names, Interests and Data Packets, and Signatures. Next to the basic architecture of an ICN forwarder, with a PIT (Pending Interest Table), a FIB (Forwarding Informationen Base), and a CS (Content Storage), the NFD has additional features like a Strategy Choice Table and a Routing Informationen Base Manager (RIB). The RIB is a usability feature for the experimental platform NFD and can handle multiple programs, users and routing protocols at the same time and merge their requests into the FIB. The Forwarding Strategy (Layer) is a part of the forwarding pipeline of NFD and decides based on the FIB whereto forward individual Interests. In detail, the Strategy Choice Table chooses for each incoming Interest a appropriate Forwarding Strategy based on prefix matching. For example an Interest of a group chat program would be forwarded by a Multicast strategy. A file download would be handled by a Forwarding Strategy, which chooses a network interface with a high throughput, whereas a metric for a telephone call would be based on a short delay.

3.2 Integration of PANINI

We integrated the behavior of PANINI directly in a Forwarding Strategy instead of as a separate program connected to the RIB. This design decision reduces the number of necessary processes and requires less computing power for our experiments. As we want to simulate networks on a single machine with over 300 nodes and several hundred producers and consumers. Furthermore, we implemented our own FIB based on a Patricia-Trie, where each edge represents a component of a name and vertices can save outgoing interfaces and replacement strategy information. This FIB is used twice in PANINI, once to save forwarding information to NACs (NAC-FIB), used for routing NAMs up the tree and once for general Interests (Interest-FIB). Our Forwarding Strategy PANINI is triggered on the receiving of an Interest. In this state the NFD has already checked that Interests are not in a loop and cannot be satisfied by the CS. Afterwards, PANINI distinguish between NAC announcements, NAMs, and general Interests. NAC announcements and NAMs are used to update the NAC-FIB and the Interest-FIB. General Interests are processed as follow. At first the Interest-FIB is checked for an available route. If a route is available, the Interest will be routed appropriately. If not, the Interest is either flooded down the tree or forwarded up to its NAC depending on its incoming interface.

To simulate in our measurements a memory scarcity, we added two polices to limit the FIB-size and replace entries in a full FIB. The first policy ads a hard limit to the FIB and performs an Least Frequently Used (LFU) algorithm to replace entries in a full FIB. The second policy adds a soft limit to the FIB, where each entry has a limited lifetime, which is renewed on usage. Both replacement strategies can be combined as well.

4 Measurement Setup

Our measurements are based on *Mininet* [8]. It enables the construction of arbitrary virtual networks on a single machine and facilitates native code running on minimal virtual nodes in these networks. We equipped Mininet based networks with our NDN/PANINI implementation. With this setup, we want to analyze the efficiency and the scalability of PANINI. In detail, we want to survey the overhead, caused by control messages and a path-stretch as a result of the required tree topology. Furthermore, we want to investigate the trade-off between FIB limitation and the flooding cost in a real-world scenario.

4.1 System Configuration

We used Mininet version 2.2.1 and NFD/NDN-cxx in version 0.4.0. All experiments are performed on *Ubuntu 14.04*, which operates on a machine with *AMD Opteron 6376* processor units (64 Cores, 500 GB main memory). Such a powerful machine was necessary, as our largest measured topology consists of over 300 virtual nodes with a NDN-Forwarder running on each as an independent process. Additionally, several hundred consumers and producers were running at the same time. For the initialization and configuration of the virtual networks, we wrote an configuration tool for Mininet in Python 2.7.6.

4.2 Network Initialization

For our experiments, we used *Uniform Recursive Trees* (URT) and *Rocketfuel* data [45] for network topologies. URTs are randomly generated trees, where each tree has the same probability to be generated. They are recursively generated as follows. The initial tree has one vertex (the root, in this case named *Host 0*) and in each generating step, a new vertex is linked to an existing vertex of the tree, whereby the existing vertex is chosen uniformly random. Rocketfuel is an ISP mapping engine, which maps traceroute data to DNS information and measures router-level ISP topologies. A dataset of six measured ISP topologies is provided on the website of the

AS	Name	#Router	Average Path Length		NAC Router
			Mesh	Tree	
1221	Telstra (au)	104	4.6	5.4	Adelaide, Australia1729
1239	Sprint (us)	315	4.0	5.2	Dallas, TX4080
1755	Ebone (eu)	87	4.5	5.5	Amsterdam, Netherlands227
3257	Tiscali (eu)	161	4.2	5.3	Frankfurt, Germany151
3967	Exodus (us)	79	4.1	5.4	Oak/Brook, IL300
6461	Abovenet (us)	138	3.8	4.6	Washington, DC483
–	URT100	100	–	6.2	Host 0

Table 4.1: Overview on used topologies. ISPs and URT with the number of (backbone) routers, the average path length of the original mesh topology and extracted shortest path tree, and the router name of the NAC.

University of Washington [46]. We constructed shortest path trees from these topologies in two steps. First, the root has been determined by the metric highest betweenness [47] (formula 4.1). The betweenness indicates the centrality of a node (v) in a network. It is calculated by counting the number of shortest paths, which traverse the node ($\sigma_{st}(v)$), divided by all shortest paths σ_{st} between node s and t. And second, the shortest path from each vertex to the root is used to create a shortest path tree. A list of all topologies is shown in table 4.1.

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{4.1}$$

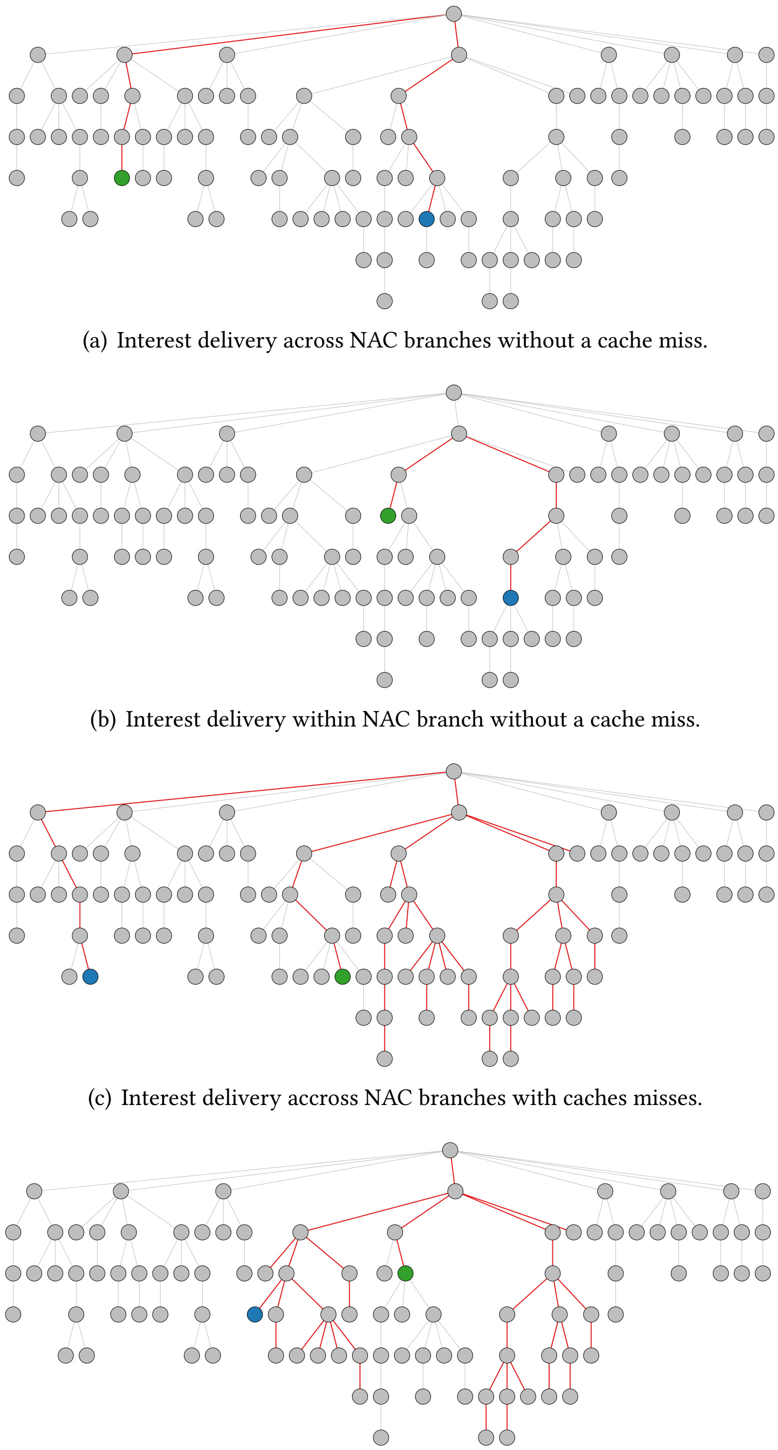
After the topologies are constructed, we placed NAC and producer in the network. The NAC is placed at the root and all producers are randomly set to a fixed position. A fixed position represents a non-mobile producer. Furthermore, we saved these positions for all upcoming measurement, to limit random effects and increase the comparability individual measurements. However, consumers do not have a fixed position, which spreads the Interest for content over the whole network.

4.3 Measurement Procedure

After the network and all parameters are chosen, an experiment can be configured and executed with the following steps. At first, Mininet is started with the chosen topology. Afterward a NFD is started on each virtual node, which is waiting for local consumers and producers on a TCP port. Furthermore, all caches are disabled to avoid related effects on our measurements. In the next step the Name Advertisement Collector (NAC) is announced in the virtual network. A dedicated Interest is flooded down the tree and each NFD saves the interface, on which the NAC message has been received, to define the upstream with the given NAC-prefix. Finally, the FIB of the NAC and of all NFDs are initialized. Each producer announces its prefix with a Name Advertisement Message (NAM). The NAM is also a dedicated Interest and is forwarded up the tree along the previously set upstream. If the network is initialized successfully, the measurement process has to handle two tasks. Task one is a maintenance thread and resends all NAMs continuously to refresh FIBs. Task two simulates the behavior of consumers and producers and is executed concurrently and sequentially. The number of concurrent processes, which we call *Interest events* in the remainder of this work, depends on the speed of the computing machine and is determined empirically. At first, a random name from the predefined name list is chosen based on its content popularity (see Chapter 2.5), which selects the traffic for each event. Then the corresponding producer is started on its predefined position in the network. Afterwards a consumer sends an Interest from random position and waits for requested data. The NFDs, the consumers, and the producer create log files on which the evaluation is based. These log files allow to extract the path each Interest has taken as well as to track the FIB-size of every NFD.

4.4 Validation Setup

Log files are used, next to the measurement evaluation, for monitoring and debugging purposes of the experiments. Figure 4.1 illustrates four examples of Interest paths measured in the *URT100* topology. This tool helps to check the PANINI implementation for proper functionality. Next the manually checks, a script checks all Interest events for packet loss, which occurs if the computing machine is overloaded. In our measurements we distinguish between two types of packet loss, (a) Interest loss, which affects our measurement, and (b) data loss, which can be neglected. Figure 4.2 summarize potential problems of a measurement of the topology *AS 1239*. The plot shows two million Interest events and is divided into *Correct events*, *Lost Interest*, *Lost Data*, and *Misc Error*. When a consumer sends an Interest and receives the requested data and no miscellaneous error occurs the event is performed correctly. An Interest is lost, when the producer does not receive the Interest. A data packet is lost when the producer receives the Interest but the consumer does not receive the data packet. Finally a miscellaneous error occurs, when an Interest has been lost in one of the broadcast branches but nevertheless reached the producer. On the one hand the Figure 4.2 shows nearly no Interest loss and no miscellaneous errors, which indicates a well performed experiment. On the other hand the figure reveals two problems we have. Problem one is the high rate of data loss on large topologies like AS 1239 and AS 3967. The simple answer would be the high computational load but contradicts the low lost rate of Interest. Problem two is the increasing rate of data loss. Our first test with larger topologies ends with an segmentation fault of the root NFD. We found out that, the segmentation fault was caused by a maintenance routine trying to clean unused faces. After disabling this routine the segmentation fault was temporary fixed but resulted in an increasing of the data loss. Our measurement results are not effected by these problems, as PANINI is not responsable for the data transport and it is not a part of our evaluation. We only have to care about the Interest lost rate, which is close to zero (usually less then ten Interest per two million events) in our measurements. Nevertheless further investigation is needed.



(a) Interest delivery across NAC branches without a cache miss.

(b) Interest delivery within NAC branch without a cache miss.

(c) Interest delivery accross NAC branches with caches misses.

(d) Interest delivery within NAC branch with caches misses.

Figure 4.1: Examples of different Interest events extracted from log files (consumer in blue, producer in green).

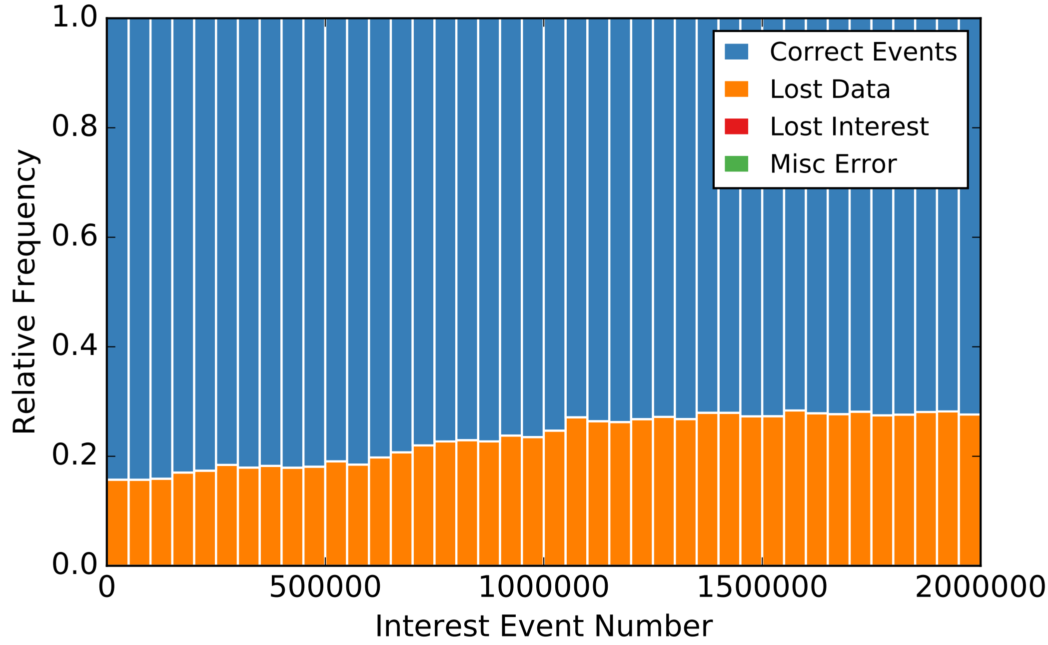


Figure 4.2: The plot summarizes potential problems of an experiment. We lost nearly no Interest, but over the measurement duration an increasing amount of data packets.

5 Evaluation

In this section, we evaluate efficiency and scalability of PANINI. These factors are depending on the overhead and routing cost. In detail, the overhead is caused by signaling messages, flooding rate, and path stretch. The routing cost is caused by the number of routing entries and its search time. We evaluate PANINI as follows. First, we investigate the impact of path stretch, which is caused by constructing tree topologies from ASes. Afterwards, we compare the FIB-size limitation with flooding costs in different AS topologies. We analyzed two simple replacement strategies for FIB entries in detail, which we integrated in our FIB implementation (see section 3.2). And finally, we survey the overall FIB-size usage, by which we can derive proposals to improve the efficiency of PANINI.

5.1 Path Stretch

PANINI operates on virtual and name-specific trees, overlayed on the physical network. These trees, on which the communication relies, can cause to a path stretch between nodes in comparison to the physical network as most Interests are forwarded over a NAC to the producer. For our investigation we compared six AS topologies with their shortest path trees with the node of the highest betweenness as root (see section 4.2). We found that the average path stretch is only between 17 and 32 percent. In most cases 50 percent of the paths do not experience any stretch at all and less than 7 percent of all paths have a stretch of more than 2. Figure 5.1 shows the cumulative distribution function for each AS. It is created by computing the length of all shortest paths between all node pairs for the origin AS network and for the constructed tree topology. Then, the quotient of the these two values have been counted. The placement of the NAC has a big influence on the path stretch. A NAC located at the center of the network minimizes the overall path stretch as many Interests traverse this node without a stretch of their paths.

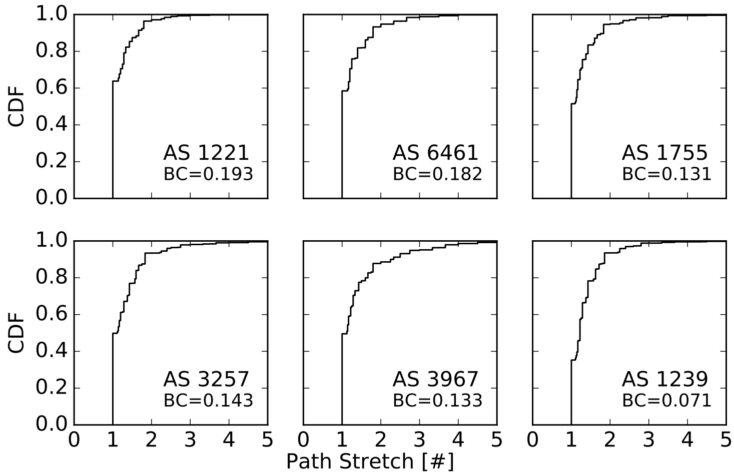


Figure 5.1: Path stretch distribution of the different Rocketfuel topologies, BC denotes the Betweenness Centrality of the NAC.

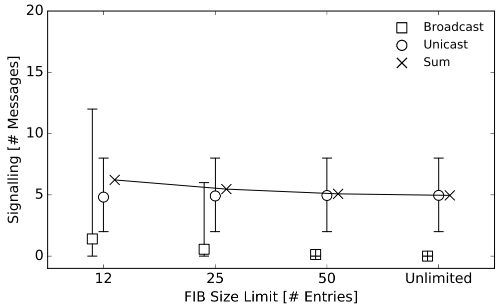
5.2 Flooding Trade-off

PANINI limits the FIB-size by dropping seldom used FIB entries. A dropped FIB entry will cause a cache miss if requested and the router is forced to flood the interest for delivery. We investigated this trade-off between FIB limitation and flooding cost and we can show that the overall broadcast rate is remarkably low. Figures 5.2 summaries our results. We focus on the ASes 1239 and 3257 and a generated URT as described in Section 4.2. For each topology we show two plots, which distinguish between the used FIB entry replacement strategy. One adds a hard limit to the FIB-size and performs a Least Frequently Used (LFU) algorithm to replace entries in a full FIB and the other adds a soft limit to the FIB, where each entry has a limited lifetime, which is renewed on usage. Each plot shows multiple measurements with a different parameter for the replacement strategy and each measurement consists of 2 Mio. Interest Events (see section 4.3) to ensure a sufficient convergence. Figure 5.2(a) for example shows the fixed FIB-size strategy with the maximum sizes 12, 25, 50 and unlimited for each router. In contrast Figure 5.2(b) shows measurements, where each FIB entry has a limited lifetime. In detail, the NAM repetition interval of producers is set to 200 seconds, which represents a tick. The Fib entry lifetime is set either to 30 seconds (0.15 ticks) or 60 seconds (0.3 ticks). Furthermore all plots show the average hop count for each measurement, divided in Unicast and Broadcast signaling. An Unicast Interest is send, when a router has a cache hit or sends the Interest to the upstream face. Broadcast Interests are send, when a router has a cache miss and has to flood the Interest down the network tree.

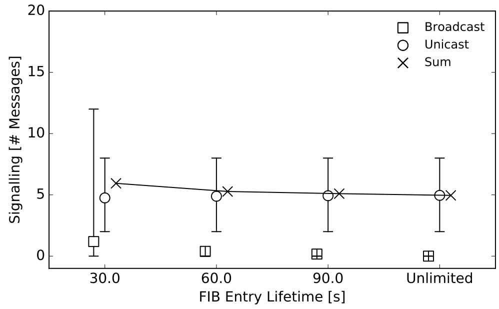
The striking point is the low number of broadcast messages per request. The average number of broadcasts are between 0.1 and 1.4, besides for the topology URT 100, which has up to 2.4 broadcast messages. These variation is caused by the different impact of the replacement strategies on the topologies. For example the URT 100 has less nodes and the NAC has less branches then AS 1239 or AS 3257. Therefore, the 1000 names we are using in our simulation have to fit in less FIBs, which is especially in the limited FIB-size replacement strategy not possible and causes to a higher flooding rate. Nevertheless, the average flooding rate is in general very low, due to the functioning of replacement strategies, which favor to drop seldom used FIB entries rather than often used once. In detail, we used the real world content popularity distribution (see section 2.5), where a very limited content set has a high request frequency and neglects seldom used content, which is flooded. In addition, we analyzed the behavior of the replacement strategies. As shown in the Figures 5.3 and 5.4 content with a high popularity are less often flooded. On the other side, these figures show also, that content of low popularity can have up to 15 broadcast packets per request and in summary with the Unicast messages over 20 packets, whereas the average path length has only 5.2 hops. But based on the content popularity distribution low ranked content has little impact on the average broadcast rate.

5.3 FIB-Size Distribution

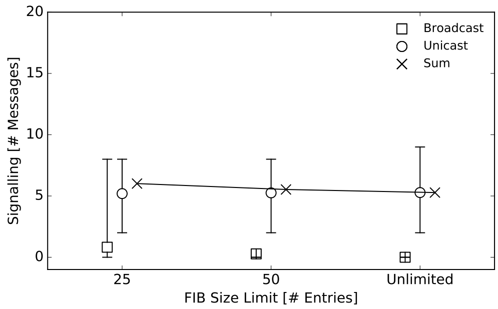
The FIB-size of a router depends on the number of local attached producers and of the number of producers in its branches. The FIB-size can be reduced by our replacement strategies, except for the NAC, which always has to hold a complete FIB. Our results show even without FIB-size reduction techniques the demanded FIB-size is moderate. For example, 90 percent of all routers in AS 1239 require 20 or less FIB entries and even 85 percent require 10 or less FIB entries by a name set of 1000 names. Figure 5.5 and Figure 5.6 show snapshots of logged FIB-sizes of all AS 1239 routers ranked by their FIB-size. In addition, all plots included the maximum possible FIB-size per router as reference, which is extracted from the measurements with unlimited FIB-size. For a sufficient convergence, all snapshots are taken after 50 percent of the executed interest events (after 1 Mio. events). The plots distinguish by the used replacement strategy. Figure 5.5 shows a plot with the replacement strategy, which adds a hard limit to the FIB-size and Figure 5.6 shows for comparison the behavior of the replacement strategy, which limits the FIB entry lifetime. The replacement strategy, which limits the maximum FIB-size, has the advantage that routers with a FIB-size below the limit do not have to drop entries, which



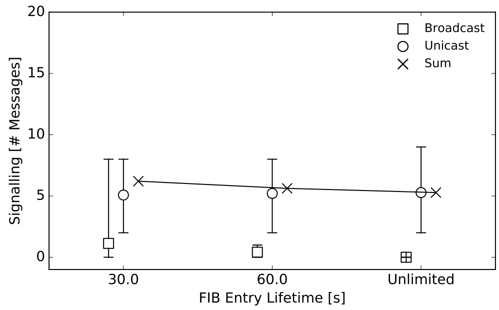
(a) AS 1239, fixed FIB-size



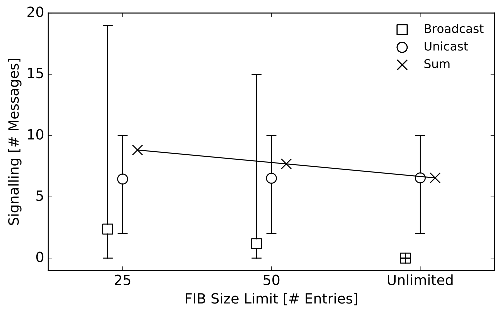
(b) AS 1239, limited FIB entry lifetime



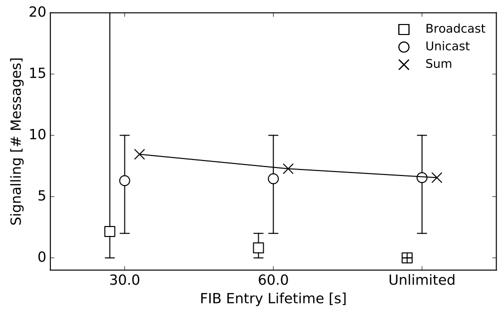
(c) AS 3257, fixed FIB-size



(d) AS 3257, limited FIB entry lifetime

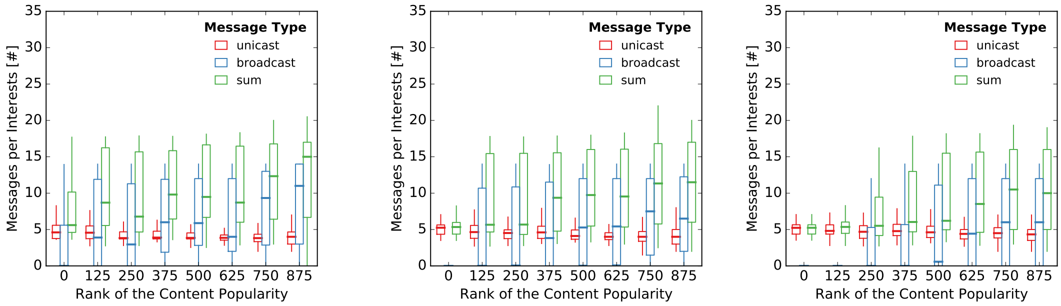


(e) URT 100, fixed FIB-size



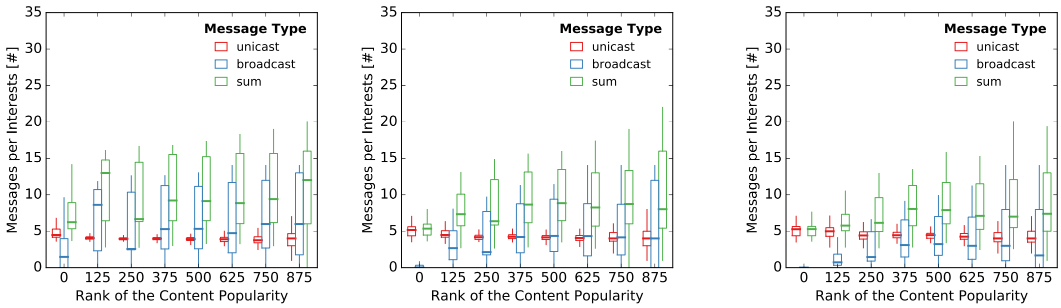
(f) URT 100, limited FIB entry lifetime

Figure 5.2: Summary of the Unicast and Broadcast Distribution for different FIB-sizes and FIB Entry Lifetimes, showing average, 95%, 5% percentile.



(a) Maximum FIB-size of 12 (b) Maximum FIB-size of 25 (c) Maximum FIB-size of 50

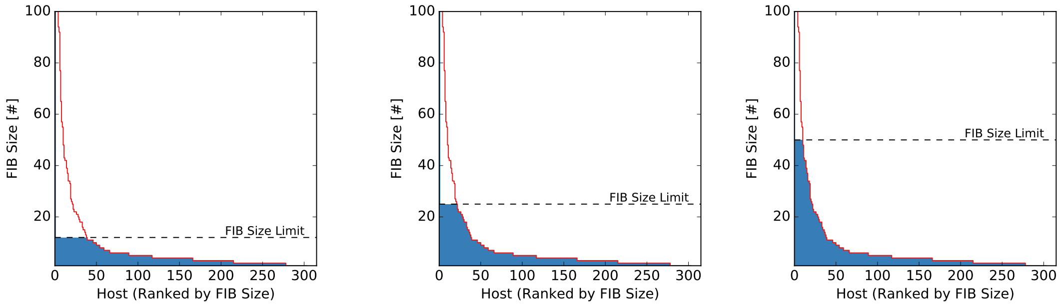
Figure 5.3: Summary of Unicast/Broadcast distribution per content for topology AS 1239 and replacement strategy limited FIB entry.



(a) FIB entry lifetime of 30 seconds (b) FIB entry lifetime of 60 seconds (c) FIB entry lifetime of 90 seconds

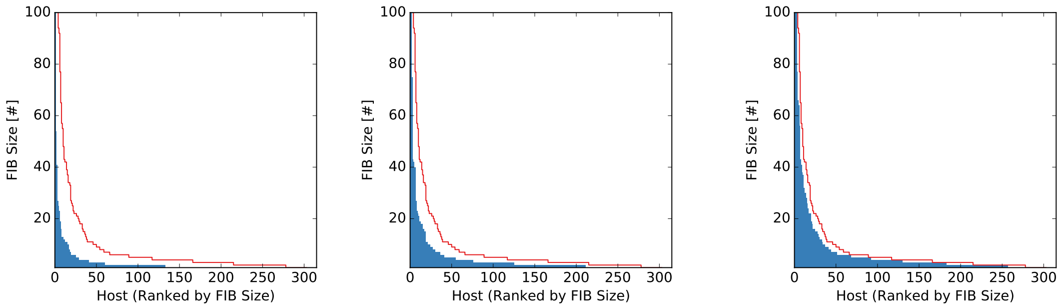
Figure 5.4: Summary of Unicast/Broadcast distribution per content for topology AS 1239 and replacement strategy limited FIB entry lifetime.

can avoid even flooding of less popular content. On the other hand, the replacement strategy, which limits the FIB entry lifetime, drops not used entries after a time interval and risks a flooding of Interests in the future.



(a) Maximum FIB-size of 12 (b) Maximum FIB-size of 25 (c) Maximum FIB-size of 50

Figure 5.5: FIB-size snapshots (blue) of AS 1239 with a limited FIB-size in comparison to a reference FIB-size of an unlimited measurement (red).



(a) FIB entry lifetime of 30 seconds (b) FIB entry lifetime of 60 seconds (c) FIB entry lifetime of 90 seconds

Figure 5.6: FIB-size snapshots (blue) of AS 2139 with limited FIB entry lifetime in comparison to a reference FIB-size of an unlimited measurement (red).

6 Conclusion and Outlook

In this work, we measured and evaluated efficiency and scalability of the name-based routing approach PANINI. We showed that the overall flooding rate is remarkably low and the FIB-size demand is (even without FIB-size reduction techniques) moderate. Additionally, we analyzed the path stretch, caused by a virtual tree topology which overlayed the physical network, is in a acceptable low range between between 17 and 32 percent.

For our measurements we integrated PANINI in the NDN-software suite. We implemented a forwarding strategy with the behavior of PANINI instead of as a separate routing program as intended by NDN. With this decision we were able to reduce the number of required processes and computation power in our measurements to enlarge our scenarios for a more realistic touch. We implemented our own FIB based on a Patricia-Trie and integrated two policies to limit the FIB-size and to simulate the handling of memory scarcity.

For our measurement scenarios we used six AS topologies measured by Rocketfuel and a topology generated by an URT algorithm. These topologies are simulated by Mininet and equipped with our NDN/PANINI implementation. In the next step, we constructed shortest path trees from these topologies and placed an aggregation point at the root node. With these simulated networks and a realistic name popularity model based on Zipf's law, which we compared to three web analytic services, we run several measurements. In summary, we could show that a well placed aggregation point in a network distributes the global FIB between ICN routers of the network and reduces the routing overhead like signaling messages and path stretch. Additionally, we could show that PANINI can limit the FIB-size of individual routers with a minimized flooding rate.

In our future work, we want to implement PANINI for the operating system RIOT [48], which is directed to low powered and memory constrained devices. Therefore, we have to focus on energy efficiency and optimized data structures for a minimal memory footprint.

Bibliography

[1] D. Kutscher, S. Eum, K. Pentikousis, I. Psaras, D. Corujo, D. Saucez, T. Schmidt, and M. Waehlisch, "Information-Centric Networking (ICN) Research Challenges," IETF, RFC 7927, July 2016.

[2] 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.

[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] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi, "A Survey on content-oriented networking for efficient content delivery," *Communications Magazine, IEEE*, vol. 49, no. 3, pp. 121–127, 2011.

[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, Poster Session*, ser. ICN 2015. New York: ACM, Oct. 2015, pp. 193–194.

[6] ——, "Let's Collect Names: How PANINI Limits FIB Tables in Name Based Routing," in *Proc. of 15th IFIP Networking Conference*. Piscataway, NJ, USA: IEEE Press, May 2016, pp. 458–466.

[7] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named Data Networking," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, 2014.

[8] "Mininet," http://mininet.org/, accessed: 2016-02-24.

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

[10] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A Data-Oriented (and beyond) Network Architecture," *SIGCOMM Computer Communications Review*, vol. 37, no. 4, pp. 181–192, 2007.

[11] 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.

[12] "The CCNx Project," http://blogs.parc.com/ccnx/, accessed: 2016-04-18.

[13] "CCN-litte Project," http://www.ccn-lite.net/, accessed: 2016-04-18.

[14] "NDN," http://named-data.net/, accessed: 2016-02-24.

[15] M. Mosko, I. Solis, E. Uzun, and C. Wood, "CCNx 1.0 Protocol Architecture," Computing Science Laborator, PARC, Tech. Rep., 2015.

[16] Z. Zhu and A. Afanasyev, "Let's ChronoSync: Decentralized Dataset State Synchronization in Named Data Networking," in *Proc. of the 21st IEEE International Conference on Network Protocols (ICNP 2013)*, 2013.

[17] M. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang, "NLSR: Named-data Link State Routing Protocol," in *3rd ACM SIGCOMM Workshop on Information-centric Networking*, ser. ICN '13. New York, NY, USA: ACM, 2013, pp. 15–20.

[18] J. Moy, "OSPF Version 2," IETF, RFC 2328, April 1998.

[19] 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

[20] ——, "Lessons from the Past: Why Data-driven States Harm Future Information-Centric Networking," in *Proc. of IFIP Networking*. Piscataway, NJ, USA: IEEE Press, 2013. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6663520

[21] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive Forwarding in Named Data Networking," *SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 3, pp. 62–67, 2012.

[22] V. Fuller and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan," IETF, RFC 4632, August 2006.

[23] H. Liu, X. De Foy, and D. Zhang, "A Multi-level DHT Routing Framework with Aggregation," ser. ICN '12. ACM, 2012, pp. 43–48.

[24] C. A. Shue and M. Gupta, "Packet Forwarding: Name-based Vs. Prefix-based," in *10th IEEE Global Internet Symp.*, 2007.

[25] "CIDR Report," http://www.cidr-report.org/as2.0/, accessed: 2016-04-16.

[26] "Hobbes Internet Timeline," http://www.zakon.org/robert/internet/timeline/, accessed: 2016-04-16.

[27] "Official Google Blog," https://googleblog.blogspot.de/2008/07/we-knew-web-was-big.html, accessed: 2016-04-18.

[28] 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.

[29] 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.

[30] D. R. Morrison, "PATRICIA – Practical Algorithm To Retrieve Information Coded in Alphanumeric," *J. ACM*, vol. 15, no. 4, pp. 514–534, 1968.

[31] H. Dai and B. Liu, "CONSERT: Constructing optimal name-based routing tables," *Computer Networks*, vol. 94, pp. 62–79, Jan. 2016.

[32] Y. Wang, T. Pan, Z. Mi, H. Dai, X. Guo, T. Zhang, B. Liu, and Q. Dong, "NameFilter: Achieving fast name lookup with low memory cost via applying two-stage Bloom filters," in *IEEE INFOCOM'13*, 2013, pp. 95–99.

[33] J. Li, J. Chen, M. Arumaithurai, X. Wang, and X. Fu, "VDR: A Virtual Domain-Based Routing Scheme for CCN," ser. ICN '15. ACM, 2015, pp. 187–188.

[34] G. Zipf, *Relative Frequency as a Determinant of Phonetic Change*. Harvard University Press, 1929.

[35] 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.

[36] M. Halvey, M. T. Keane, and B. Smyth, "Mobile web surfing is the same as web surfing," *Communications of the ACM*, vol. 49, no. 3, pp. 76–81, 2006.

[37] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System," in *7th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '07. ACM, 2007, pp. 1–14.

[38] G. Dán and N. Carlsson, "Power-law Revisited: Large Scale Measurement Study of P2P Content Popularity," in *9th International Conference on Peer-to-peer Systems*, ser. IPTPS'10. USENIX Association, 2010, pp. 12–12.

[39] L. Saino, I. Psaras, and G. Pavlou, "Hash-routing Schemes for Information Centric Networking," in *Proceedings of the 3rd ACM SIGCOMM Workshop on Information-centric Networking*, ser. ICN '13. New York, NY, USA: ACM, 2013, pp. 27–32.

[40] C. Holmberg, S. Hakansson, and G. Eriksson, "Web Real-Time Communication Use Cases and Requirements," IETF, RFC 7478, March 2015.

[41] C. Vogt, M. J. Werner, and T. C. Schmidt, "Leveraging WebRTC for P2P Content Distribution in Web Browsers," in *21st IEEE Int. Conf. on Network Protocols (ICNP 2013), Demo Session*. Piscataway, NJ, USA: IEEE Press, Oct. 2013, ICNP Best Demo Award.

[42] R. Ravindran, A. Chakraborti, and aytav.azgin, "Forwarding-Label support in CCN Protocol," IETF, Internet-Draft – work in progress 02, March 2016.

[43] "NFD Developer's Guide," http://named-data.net/publications/techreports/ndn-0021-6-nfd-developer-guide/, accessed: 2016-04-03.

[44] "NDN Protocol," http://named-data.net/doc/ndn-tlv/, accessed: 2016-03-31.

[45] 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.

[46] "Rocketfuel Data," http://research.cs.washington.edu/networking/rocketfuel/, accessed: 2016-02-24.

[47] L. C. Freeman, "A Set of Measures of Centrality Based on Betweenness," *Sociometry*, vol. 40, pp. 35–41, 1977.

[48] E. Baccelli, O. Hahm, M. Günes, M. Wählisch, and T. C. Schmidt, "RIOT OS: Towards an OS for the Internet of Things," in *Proc. of the 32nd IEEE INFOCOM. Poster*. Piscataway, NJ, USA: IEEE Press, 2013.