Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia

COPPE UFRJ

AN AUTONOMOUS AND EFFICIENT CONTROLLER-BASED ROUTING SCHEME FOR NETWORKING NAMED-DATA MOBILITY

João Vitor Torres

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Elétrica.

Orientador: Otto Carlos Muniz Bandeira Duarte

Rio de Janeiro
Junho de 2016

AN AUTONOMOUS AND EFFICIENT CONTROLLER-BASED ROUTING SCHEME FOR NETWORKING NAMED-DATA MOBILITY

João Vitor Torres

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Otto Carlos Muniz Bandeira Duarte, Dr.Ing.

Prof. Lisandro Zambenedetti Granville, Ph.D.

Prof. Artur Ziviani, Dr.

Prof. Igor Monteiro Moraes, D.Sc.

Prof. Miguel Elias Mitre Campista, D.Sc.

Prof. Pedro Braconnot Velloso, Dr.

RIO DE JANEIRO, RJ – BRASIL
JUNHO DE 2016

Torres, João Vitor

An Autonomous and Efficient Controller-based Routing Scheme for Networking Named-Data Mobility/João Vitor Torres. – Rio de Janeiro: UFRJ/COPPE, 2016.

XVII, 83 p.: il.; $29,7$cm.

Orientador: Otto Carlos Muniz Bandeira Duarte

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2016.

Referências Bibliográficas: p. 67 – 76.

1. Redes Orientadas a Conteúdo. 2. Redes Definidas por Software. 3. Separação de Planos de Dados e Controle. 4. Mobilidade. I. Duarte, Otto Carlos Muniz Bandeira. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

À minha família.

Agradecimentos

A realização deste trabalho contou com o suporte fundamental de muitas pessoas, tanto em aspectos técnicos, quanto em incentivo diante de momentos desafiadores. Desta forma, agradeço:

Aos meus pais, Valdete e Edílio, e a toda minha família pelo apoio e incentivo contínuo à minha formação.

À minha esposa Claudia pelo companheirismo e compreensão.

Ao professor Otto, meu orientador, pela oportunidade, pela orientação, pelo incentivo, pelas críticas, pelo desafio, pela busca do autoconhecimento, pelo esforço, pela paciência e pela amizade construída no período deste trabalho.

Aos professores Lisandro Zambenedetti Granville, Artur Ziviani, Igor Monteiro Moraes, Miguel Elias Mitre Campista e Pedro Braconnot Velloso pelas sugestões na qualificação e pela participação na banca examinadora.

Ao professor Raouf Boutaba pelas sugestões ao trabalho.

A todos os amigos do GTA, em especial Lyno, Martin, Diogo e Igor, pelos conselhos e pela grande ajuda.

Aos colegas de trabalho da PETROBRAS pelo incentivo e apoio.

Aos funcionários do Programa de Engenharia Elétrica da COPPE/UFRJ, Daniele, Maurício e Rosa pela presteza no atendimento na secretaria do Programa.

Por fim, agradeço a PETROBRAS, FINEP, CNPq, CAPES, FAPERJ e UOL pelo financiamento deste trabalho.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

UM ESQUEMA BASEADO EM CONTROLADOR PARA ROTEAMENTO AUTÔNOMO E EFICIENTE EM REDES ORIENTADAS A CONTEÚDO COM MOBILIDADE

João Vitor Torres

Junho/2016

Orientador: Otto Carlos Muniz Bandeira Duarte

Programa: Engenharia Elétrica

A enorme quantidade de dados disponível em Redes Orientadas a Conteúdo desafia o tamanho das tabelas de rotas e as técnicas para localização e encaminhamento de informação. A mobilidade e as cópias de conteúdo em diferentes localizações agravam o desafio de escalabilidade. Esta tese propõe e analisa o desempenho de um esquema específico de roteamento baseado em controlador, chamado CRoS-NDN (Controller-based Routing Scheme for Named-Data Networking), que preserva todas as funcionalidades das Redes Orientadas a Conteúdo usando apenas pacotes de Interesse e Dados. O esquema proposto suporta a mobilidade e provê a rápida recuperação do conteúdo a partir de cópias fora do caminho entre consumidor e produtor, pois separa identificação e localização sem explodir as tabelas de rotas ou supor agregação de prefixos. O CRoS-NDN provê funcionalidades similares às redes par-a-par e de distribuição de conteúdo em redes NDN e melhora a eficiência para conteúdos populares. São definidos nomes e procedimentos específicos para a comunicação de roteadores e controlador. O CRoS-NDN evita a sobrecarga de mensagens de controle codificando informação de sinalização nos nomes de conteúdo, os quais disparam ações dos roteadores. Adicionalmente, o esquema proposto evita a replicação de informação de roteamento do controlador para os roteadores porque estes solicitam rotas por demanda. A proposta deste trabalho é comparada com protocolos distribuídos de roteamento. São derivadas expressões analíticas para o limite inferior da eficiência e o limite superior do atraso. Os resultados de simulação para a eficiência e para o atraso mostram que o esquema proposto é robusto para uma ampla gama de cenários. Além disso, o esquema economiza recursos computacionais para uma quantidade crescente de prefixos.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

AN AUTONOMOUS AND EFFICIENT CONTROLLER-BASED ROUTING SCHEME FOR NETWORKING NAMED-DATA MOBILITY

João Vitor Torres

June/2016

Advisor: Otto Carlos Muniz Bandeira Duarte

Department: Electrical Engineering

The huge amount of content names available in Named-Data Networking (NDN) challenges both the required routing table size and the techniques for locating and forwarding information. Content copies in different locations and content mobility worsen the scalability challenge. We propose and analyze the performance of a specific Controller-based Routing Scheme, named CRoS-NDN, which preserves all NDN features using the same interest and data packets. The proposed scheme supports content mobility and provides fast content recovery from copies that do not belong to the consumer-producer path because it splits identity from localization without incurring routing table explosion or supposing prefix aggregation. CRoS-NDN provides features similar to peer-to-peer and Content Distribution Network (CDN) in NDN, and it improves the efficiency for popular content. We define specific names and specific procedures for routers and controller efficient communication over NDN. CRoS-NDN adds router actions and avoids control message overhead by coding signaling information on content names. Additionally, our scheme avoids the replications of routing information from controller to routers because they request the routes on-demand. We compare our proposal with other distributed routing protocols. We derive analytical expressions for lower-bound efficiency and for upper-bound latency. In addition, we provide simulation results for data delivery efficiency and data delivery latency. The simulation results show the proposed scheme is robust for a wide range of scenarios. Furthermore, CRoS-NDN shows an economical use of computational resources for a growing number of prefixes.

Contents

List of Figures

List of Tables

List of Symbols

AD	**A**nnouncement **D**elay, p. 31
AP	**A**nnounced **P**refixes, p. 31
AR	**A**nnouncement **R**ate, p. 31
CD	**C**onsumer-producer **D**elay, p. 31
CR	**C**onsumer **R**ate, p. 31
DDD	**D**ata **D**elivery **D**elay, p. 31
DDE	**D**ata **D**elivery **E**fficiency, p. 31
FF	**F**IB match **F**ail ratio, p. 31
H	Network diameter **H**ops, p. 31
KR	**K**eepalive **R**ate, p. 31
L	Number of **L**inks, p. 31
LD	**L**ink **D**elay, p. 31
N	Number of **N**odes, p. 31
RTD	Max **R**ound **T**rip **D**elay, p. 31
S_i	State i, p. 8
TD	**T**opology-update **D**elay, p. 31
TR	**T**opology change **R**ate, p. 31
T_i	Transition i, p. 8

List of Abbreviations

ARP	Address Resolution Protocol, p. 25
BGP	Border Gateway Protocol, p. 2
CRoS-NDN	Controller-based Routing Scheme for Named-Data Networking, p. 5
CS	Content Store, p. 5
DDD	Data Delivery Delay, p. 30
DDE	Data Delivery Efficiency, p. 30
DNS	Domain Name System, p. 6
FIB	Forwarding Information Base, p. 2
ICN	Information-Centric Network, p. 3
ID	Identifier, p. 12
LSA	Link State Advertisement, p. 27
LSDB	Link State DataBase, p. 27
NDN	Named-Data Networking, p. 1
NLSR	Named-Data Link State Routing, p. 27
OSPF	Open Shortest Path First, p. 2
PIT	Pending Interest Table, p. 5
RIP	Routing Information Protocol, p. 26
SDL	Specification and Description Language, p. 13
SDN	Software Defined Network, p. 5
ndnSIM	NS-3 based Named-Data Networking (NDN) simulator, p. 7

Chapter 1

Introduction

The Internet Protocol (IP) [1] was designed to establish conversational communication between host pairs. The network forwards packets based on host identifiers and IP routers have no visibility of packet content. Therefore, the network cannot optimize the link usage reducing the exchange of redundant content in parallel conversations. This imposes a scalability challenge to provide access to high popular content in flash crowd events like, for example, the Olympic Games. The huge number of network users, which is reaching all the population [2, 3], increases both the frequency and the intensity of flash crowd events. In addition, IP applications must resolve the identifier of the host that stores the desired content and this identifier depends on the host location. In consequence, the host mobility imposes another challenge because the change of host identifiers breaks an ongoing conversation. The increasing number of mobile devices, which is also reaching all the population [2, 3], exacerbates the issue.

Named-Data Networking (NDN) applications refer directly to content names, avoiding host network identifiers for communication [4]. In this new paradigm, both the host mobility/multihoming and the content mobility/multihoming do not concern applications. NDN network layer focuses on unique network-visible names that identify content. This network layer forwards two types of packets: the interest and the data packets. The interest packet expresses consumers will for content and leaves breadcrumbs on each hop to reach the consumer back. Hence, for each interest packet, the network replies with a data packet containing the desired content. The NDN ensures efficient communication, load balance, energy efficiency, and flow control through popular content storage and data packet replies from any content cache copy [4–7]. In addition, NDN is incrementally deployable because NDN packets can be transported over Internet Protocol (IP) or can replace IP. More importantly, interest and data packets one-to-one correspondence avoids link congestion due to Distributed Denial-of-Service (DDoS) attacks. NDN routers[1] aggregate interests for

[1]The word router refers to a content router, and there is no distinction between Named-Data

the same content and limit the amount of unanswered interests [9]. Furthermore, unlike IP Multicast, NDN flow control is receiver-oriented and adapts to the link capacity of each individual consumer.

Named-data routers find and deliver content based on its name. Therefore, NDN routing schemes announce named-data prefixes diffusing their associated data location. NDN routing schemes based on Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP) inherit IP characteristics due to their focus on prefix dissemination and routing [4]. These routing schemes suffer with the amount of named-data prefixes that is intrinsically higher than the required IP prefixes. In addition, in order to reach content copies stored outside their original locations due to mobility, multihoming, and cache, NDN announces more routes with less-aggregated prefixes. In these scenarios, the routing schemes should store more routes and exchange more control messages to announce all the addressable content, which results in high control overhead and possible risk of Forwarding Information Base (FIB) explosion [10]. On the other hand, announcement suppression of non-aggregated prefixes reduces the cache-hit opportunities just to copies located along the path from consumer to producer [11]. Caching along the path supposes that router caches are big enough to accommodate popular contents that last enough time to respond repeated requests. This is a technical and economical trade off considering the amount of available content and the long tail for the content popularity distribution [12]. The long-tail popularity distribution of content implies that the most of content items have similar request probability and, therefore, the cache hit probability is low for reduced cache sizes compared to the amount of content.

In Figure 1.1, we illustrate the scenario that routers forward the consumer interest straight to producer. A content copy for the interest is available at a closer host, but routers do not have forwarding rules to reach this copy. The forwarding rules employ the prefix of the content name and the prefix `/producerprefix` for the content name `/producerprefix/wantedcontentsuffix` points just the direction to the producer. In order to turn the content copy reachable, routing schemes based on prefix announcements need to add a more specific forwarding rule with the prefix for the specific content `/producerprefix/wantedcontentsuffix`. The higher is the number of prefixes, the higher is the control message overhead of these routing schemes, and the higher is the required memory for the router table of forwarding rules.

Networking and Content-Centric Networking [8].

Table of Forwarding Rules		
Prefix	**Destination**	**Status**
/producerprefix	Producer	Active
/producerprefix/wantedcontentsuffix	Host	Missing

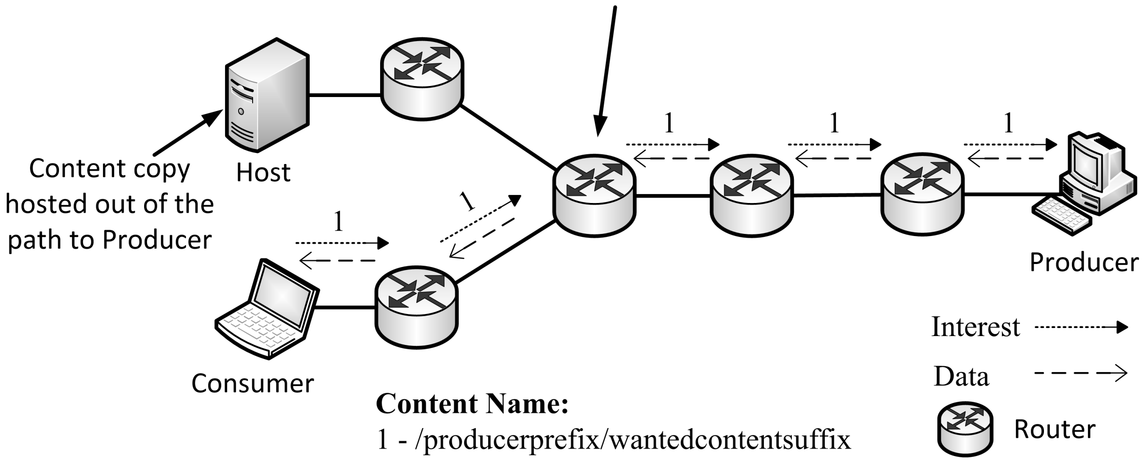


Figure 1.1: Consumer interest forwarding and data retrieval from content hosted at the producer. NDN routers do not reach the closer content copy hosted out of the path to Producer because the location of the content copy is not announced by the routing protocol.

1.1 Related Work

Ghodsi et al. discourage Information-Centric Network (ICN) research due to the very long tail of content popularity distribution [13]. They argue that pervasive cache at all routers is worthless for an approach that cache only along the path to producer and that a single proxy cache would provide the same results. We observe that NDN mitigates server load in flash crowd events and Distributed Denial-of-Service (DDoS) attacks that are not solved by a single proxy cache. In addition, they argue that locating content copies outside the path to producer requires a localization resolution system that works at the rate given by the ratio of packet speed to mean object size. We note that the very long tail stands for aggregated measures of content popularity distribution taken for thousands of consumers employing large time windows. On the other hand, individual consumers present a much less flatter tail for popularity distribution measures of content prefixes taken for smaller time windows [14, 15]. Thus, we argue that access routers cache the localization resolution data for local consumers. Additionally, the volume of video traffic dominates the total IP traffic today and keeps growing [2, 3]. The video traffic contributes to a lower rate of localization requests due to the large content size. Therefore, like proposals [16–19], we argue that locating content copies outside the path to producer is worthy.

Various aspects of Information-Centric Network (ICN) research are presented in surveys and all of them point scalability as a major challenge [20–31]. We argue that our proposed routing scheme reduces the routers memory requirement and the number of control messages pointed as a scalability challenge due to the vast size of the content naming space.

A number of schemes address content network, but propose a publisher-subscriber architecture [32–34]. We consider publish-subscribe approach is vulnerable to denial of service attacks, because it does not preserve the packet flow balance provided by on demand approach for individual data packets. Other schemes address the mapping problem of content identifier to location [35–40]. For example, Baid *et al.* propose a two level indirection scheme that maps named-data prefixes to a reduced set of flat identifiers and, then, these identifiers into network addresses [38]. The Baid *et al.* scheme employs a distributed hash tables (DHT) system to provide this indirection that reduces the FIB memory requirement and the message exchange, but, like the cited mapping schemes, it does not preserve content names on forwarding decisions. We argue that our scheme can be extended to incorporate a scalable resolution scheme to execute this mapping; however, the extension should preserve the content name orientation on packet forwarding decisions to maintain the aggregation/caching opportunities and to adapt the forwarding plane to data mobility.

Afanasyev *et al.* propose a Domain Name System (DNS) to map and encapsulate data names in a reduced set of network names related to network domains [37]. The scheme reduces the FIB memory requirement, however, DNS servers have no clue of the request originator and, thus, DNS response contains multiple names and routers must execute multiple prefix-based lookups to find the shortest path choice for each content. They argue that name changes must be avoided due to complex implications on the named-based scheme. Zhang *et al.* propose a tunneling approach that changes content name and inherits the NDN benefits. We argue that both approaches should be further investigated and, more importantly, these two proposals are orthogonal to our Controller-based Routing Scheme for Named-Data Networking (CRoS-NDN) and one can be integrated to CRoS-NDN providing higher scalability on content location storage and retrieval.

A number of schemes propose Software Defined Network (SDN) technology to consolidate routing information on a centralized controller [41–44]. Fernandes *et al.* observe controller-based solutions alleviate general packet forwarding nodes from control message processing and fit well for next generation networks [45]. Rothenberg *et al.* argue the controller single point of failure is in general redundant and each controller takes charge for a limited subset of nodes overcoming the centralized criticism [46]. Shi *et al.* propose a data synchronization scheme for NDN that can replicate the controller information [47] and provide redundancy. Gao *et al.* proposes a scalable area-based hierarchical architecture (SAHA) for intra-domain communication to address the control plane scalability problem [48]. Salsamo *et al.* propose the OpenFlow-based architecture for the SDN technology applied to ICN [49]; however, the OpenFlow approach brings the well-known IP restrictions, for example, host mobility and multihoming [50]. We argue that the software-defined network approach overcomes the unnecessary control message flooding and reduces the router FIB memory requirement by storing only active consumed prefixes instead of all published prefixes, which is orders of magnitude higher than the active consumed prefixes [51], and by replacing the oldest added routing rules with new ones. We also argue that the on-demand route-request avoids the replications of routing information from controller to routers upon topology change or content mobility. In addition, the routers and the controller may sign the interests for security provenance and validity, as in VoCCN [52].

1.2 Controller-based Routing Scheme Proposal

We propose the Controller-based Routing Scheme for Named-Data Networking (CRoS-NDN) that follows the Software Defined Networks (SDN) technology and preserves the same interest and data packets defined by Named-Data Networking (NDN) [53, 54]. Our proposal does not require additional packets. Therefore, packet forwarding follows default router processing through Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB) as detailed in [6]. Consequently, it preserves NDN features such as congestion control, network failure detection, and path diversity. Like OpenFlow-based solutions for Information-Centric Networks (ICN), CRoS-NDN consolidates the control plane on the controller, which is responsible for the named-data location storage and routing, but employs only NDN packets for router-controller communication. Thus, CRoS-NDN avoids IP restrictions on host mobility and multihoming. The controller acquires the network topology in a bootstrap phase and calculates routes to all routers. Then, the router-to-controller routes are installed in all routers. After the bootstrap phase, the routers register the named-data location in the controller on behalf of connected producers and, on its turn, the controller stores the location of all registered named-data and calculates routes to every valid named-data. Hence, a router requests the controller for a new route to any unknown prefix.

Our proposal CRoS-NDN splits the content names from the content localization and it forwards interests to the closest registered copy, irrespectively its location outside the path to producer. Unlike routing schemes based on prefix announcements, CRoS-NDN does not impose hierarchically indexed prefixes tied to location in order to summarize routing information that must fit in FIB size, neither incurs FIB size explosion. CRoS-NDN routers forward interest based on content names and the controller evaluates routes based on content location in network topology. However, unlike the location resolution approach of Domain Name System (DNS) [2], CRoS-NDN localization is topology aware. Our scheme caches, closer to consumers, content copies that are less popular than the ones cached on routers along the path to producer. Therefore, CRoS-NDN provides low latency features like Content Distribution Networks (CDNs) and peer-to-peer applications. Additionally, CRoS-NDN automatically discovers/configures routers and controller and, thus, it avoids manual provisioning. In addition, this automation introduces low control overhead because it restricts the interest flooding to specific name prefixes employed for routers and controller auto discovery. Furthermore, CRoS-NDN improves the mobility efficiency of content and content host because our scheme consolidates the routing information for content localization and for router adjacencies. This consolidation, provided by our controller-based approach, allows the usage of known technics to supply elastic resources for controller computation of routes and storage of content locations employing well-connected data center infrastructures. Unlike CRoS-NDN consolidation of controller functions, distributed approaches require the design of routers with processing power capacity and storage space for peak-utilization events of its local control plane functions. These peak-utilization events occur during network changes, while, most of the time, routers run with spare resources in distributed approaches [56].

CRoS-NDN reduces the router-controller communication overhead to fewer messages. Our scheme adds router actions and avoids control message overhead by coding signaling information on content names. Additionally, CRoS-NDN installs a new route on all routers in a path to content with a single route request to controller. The route-requester router directly instructs the new forwarding rule to routers in the path to content and this instruction avoids new requests to controller. Furthermore, CRoS-NDN avoids the constant replication of routing information from controller plane to routers data plane. The CRoS-NDN router updates the forwarding plane by requesting new routes to controller upon no-response time-expiration of interests. Thus, the scheme reduces the overhead of communication between routers and controller from the large number of available prefixes to the fraction of consumer momently-requested prefixes. Moreover, CRoS-NDN reduces the router FIB memory requirement by storing only active consumed prefixes instead of all published prefixes and by replacing the oldest added routing rules with new ones.

We analyze the CRoS-NDN efficiency considering a single administrative domain and compare the results with other known distributed schemes. Our evaluation measures the communication overhead and the data delivery latency of each

[2] Rula *et al.* studies the DNS effectiveness as a consumer localization method. They conclude DNS is unsuitable for content consumer localization in modern cellular networks [55].

scheme. We derive expressions for lower bounds of the communication efficiency and upper bounds for the latency, worst-case scenario. We implement our proposal and the other distributed protocol in the ndnSIM [57] simulator and we run a set of simulations to compare the different approaches. The obtained results demonstrate that our proposal CRoS-NDN is robust in relation to the number of prefixes when compared with the distributed schemes and improves the efficient for the mobility of content producers.

The rest of this thesis is structured as follows. Chapter 2 presents the CRoS-NDN proposal, the protocol messages, message time sequence, and main features considering a single administrative domain. Chapter 3 describes the comparing distributed routing schemes. Chapter 4 presents the performance analysis of each scheme considering the convergence delay and the data delivery efficiency metrics. Chapter 5 presents the simulation environment and the set of simulation results that jointly evaluates the scheme performance and demonstrates the scheme properties. Finally, Chapter 6 concludes and presents future research directions.

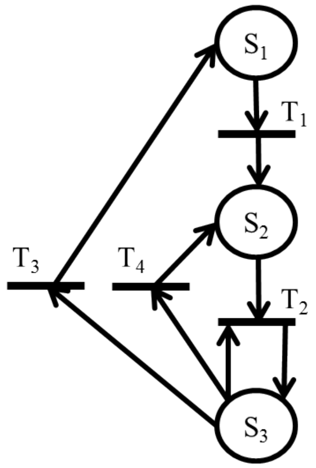
Chapter 2

The Proposed Routing Scheme: CRoS-NDN

Our Controller-based Routing Scheme for Named-Data Networking (CRoS-NDN) is composed of two phases: the Bootstrap phase and the Named-Data Routing phase. The Bootstrap phase monitors router adjacencies and assures the knowledge of the global network topology. The Named-Data Routing phase guarantees the localization and access to the requested content. We consider two network elements: one controller and routers. Routers forward packets to destination, cache content, and register the named-data location on behalf of producers. Moreover, routers request to the controller paths for unknown content names. The controller calculates routes and stores named-data locations. The control plane consolidation ensures an efficient way to register and retrieve content location without flooding the entire network. We consider a network of a single administrative domain.

Routers proactively register network information on the controller and they reactively request new routes to the controller upon consumer interests to locally unknown name prefixes. Figure 2.1 shows the state transition diagram for the router-controller interaction. A router starts at S_1 state searching a controller and changes to state S_2, transition T_1, whenever the router finds a controller. At S_2, when required, the router sends requests to the controller, transition T_2, changes to state S_3, and waits for the response. At S_3, when required, the router can send new requests to the controller without changing its state. Whenever a router-controller request expires without response, transition T_3, the router state changes to S_1. At S_3, the router can move to S_2 when the controller answer all pending requests, transition T_4.

The proposed scheme autonomously finds a path from every router to the controller. In other words, CRoS-NDN does not require a preexisting direct physical or logical connection between routers and the controller. This important feature preserves the original NDN stack and, unlike OpenFlow-based solutions, removes



State / Transition – Description
S_1 - Router searching for controller
S_2 - Router with an active controller
S_3 - Router awaiting controller response
T_1 - Router finds the controller
T_2 - Router sends a request to controller (router registration, named-data registration, or route request)
T_3 - Router-controller request expires without response
T_4 - Router receives all pending controller responses

Figure 2.1: The state transition diagram showing router interaction with the controller.

IP dependencies for ICN routers and controller communication. Hence, CRoS-NDN automates the configuration of routers and controller and, thus, it avoids manual provisioning of network routers and IP addresses. Our scheme reduces routing signaling overhead by restricting network interest flooding. Routers only flood the network to initially find the controller, during the Controller Discovery procedure. Afterwards, the controller discovery only repeats upon no-response time expiration of router to controller interest. Furthermore, cache and interest aggregation reduce the discovery overhead. Therefore, CRoS-NDN wider broadcast domain does not incur additional signaling overhead for controller discovery [1]. Each router monitors its one-hop neighbors, by Hello procedure, and the router registers any topology change in the controller, during Router Registration procedure. Routers also register in the controller the name prefixes of local produced content, Named-Data Registration procedure. The controller stores the received information from network routers and it acquires knowledge of the network topology and of content location.

Unlike OpenFlow-based solutions that each router in consumer-producer path requests the controller a route, CRoS-NDN end-to-end route installation charges the controller with only one route request, during Route Request procedure. The route-requesting router informs its identifier and the requested content name in the route request sent to the controller. Upon the route request, the controller identifies the requesting router and, then, it locates the content producer router. Afterwards, the controller computes the sequence of router identifiers in the path from consumer to producer and, then, the controller answers the route request. Upon route-request

[1] In order to deploy CRoS-NDN over IP, we note that IP Multicast is a solution to reach multiple IP subnets in a single domain and find the controller. Unlike OpenFlow, CRoS-NDN does not require manual configuration of routers with the controller IP address that must be reachable a priori.

controller answer, the requesting router builds a specific interest that installs the new FIB entry on each router in the path from consumer to content producer, Route Installation procedure. Although the path calculation relies on router identifiers, the content-request interest forwarding relies only on content names.

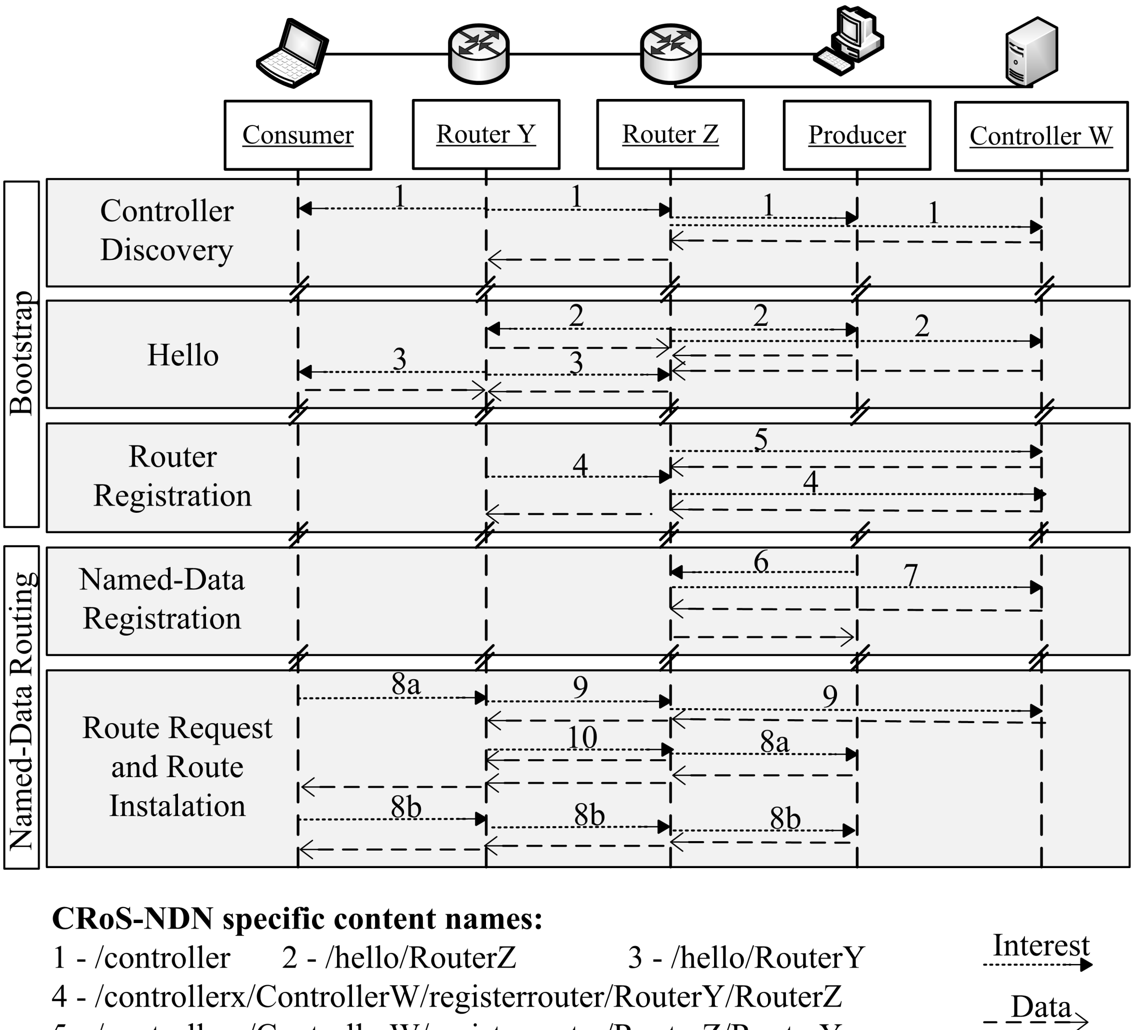
Our scheme natively splits content identity from content localization, enabling content mobility. The Named-Data Registration procedure provides content-copies reachability at any location. In addition, the Route Request procedure jointly resolves the content location and evaluates the best route from consumer to the content copy with the lowest cost. Therefore, CDN servers store content copies and register content location in the controller. Likewise, peer-to-peer application nodes at any location register content names to cooperate directly among themselves.

Topology changes or content mobility can invalidate FIB router entries. Therefore, unlike OpenFlow-based solutions that the controller proactively updates all routers FIB upon any change [2], CRoS-NDN router employs a data-plane feedback procedure to remove invalid entries from local FIB. Interests without response cause Pending Interest Table (PIT) entry removal after the interest lifetime expiration. Then, on PIT entry removal, our scheme erases the associated FIB entries. It is of utmost importance to note that CRoS-NDN scheme employs local router state, PIT, to remove invalid routes only on routers actively using that FIB entry and our scheme avoids the replication of route information from controller to routers. PIT expiration is native in NDN, but CRoS-NDN adds specific actions to remove invalid forwarding rules in the Forwarding Information Base (FIB) upon PIT entries expiration. Furthermore, CRoS-NDN routers update the controller topology view upon failure to reach neighbor routers. Unlike NDN, CRoS-NDN establishes how to feedback network changes to the routing protocol based on PIT entry expiration. In addition to reduce the signaling overhead, CRoS-NDN lessens the requirement for FIB memory router to the scale of simultaneous consumed prefixes. Our scheme reuses FIB memory and replaces old entries with new ones. This is in contrast to supporting all content prefixes available on the network irrespectively of consumer pattern of content requests for different prefixes.

CRoS-NDN executes the Controller Discovery, Hello, and Router Registration procedures in the Bootstrap phase. In Named-Data Routing phase, our scheme executes the Named-Data Registration, Route Request, and Route Installation procedures. Figure 2.2 presents the interest and data sequence of our scheme procedures.

We define specific names and specific procedures for routers and controller efficient communication over NDN. Therefore, CRoS-NDN preserves NDN features

[2]OpenFlow can install new forwarding rules reactively or proactively; however, the OpenFlow installed rules must be updated proactively upon topology changes to avoid forwarding loops and black holes.



CRoS-NDN specific content names:

1 - /controller 2 - /hello/RouterZ 3 - /hello/RouterY

4 - /controllerx/ControllerW/registerrouter/RouterY/RouterZ

5 - /controllerx/ControllerW/registerrouter/RouterZ/RouterY

6 - /registerNamedData/myprefix

7 - /controllerx/ControllerW/registerNamedData/RouterZ/myprefix

8a - /myprefix/seq1 8b - /myprefix/seq2

9 - /controllerx/ControllerW/routeFrom/RouterY/myprefix/seq1

10 - /router/RouterZ/installRouteAndForward/RouterZ/prefixSize/myprefix/seq1

Figure 2.2: The Interest/Data packet time sequence for CRoS-NDN procedures. (1) Routers Y and Z find Controller W by sending a controller discovery message. (2) and (3) Routers Y and Z send a hello message to inform each other their presence. (4) and (5) Each router sends a router register message to register its neighbors in Controller W. (6) and (7) The Producer sends a named-data registration message to Router Z to register a named-data, in controller W. (8a) and (9) Consumer requests a content, sending a content-request message, and Router Y requests Controller W a new route for the named-data. (10) Router Y requests Router Z to install a new route to the named-data. (8b) Routers Y and Z forward further Consumer interests directly to Producer.

keeping the named-data packet-forwarding scheme of NDN. In other words, unlike OpenFlow-based solutions, our proposal removes the dependency on IP for routers communication with a consolidated control plane. CRoS-NDN expands the default processing by adding router specific actions based on specific data names. Our CRoS-NDN proposal codifies the signaling information on specific data names, avoiding control message overhead. Therefore, CRoS-NDN sends the routing information embedded in content names similarly to Jacobson *et al.* strategy applied on SIP (Session Initiation Protocol) invite messages for Voice over CCN (VoCCN) [52].

In this thesis, we refer as named-data to any addressable and reachable data such as file, services, or network elements. All routers and controllers own a unique identification (ID), and, then, they are also addressable in the network. We define five specific data name prefixes reserved for the routing scheme: `/hello`, `/router`, `/controller`, `/controllerx`, and `/registerNamedData`. An interest with the data name `/hello` prefix followed by the router ID advertises the router presence to its neighbors; an interest with the data name `/router` prefix followed by the router ID addresses a specific router; an interest with the data name `/controller` prefix addresses any controller; an interest with the data name `/controllerx` prefix followed by the controller ID addresses the controller; and, finally, an interest with the data name `/registerNamedData` prefix requests the registration of new named-data.

Routers start without any forwarding rule in FIB, except the forwarding rules or procedures that the routers themselves process such as: `/hello`, `/hello/routerID`, `/controller`, and `/registerNamedData`. FIB entry `/hello` points to the router internal application that processes neighbor keep-alive messages. FIB entries `/hello/routerID` and `/controller` point to all neighbor interfaces. The `/registerNamedData` FIB entry points to the router internal application that processes named-data registration requests from users. Based on the defined initial rules, the routers initiate the bootstrap phase to enable the controller route computation on the named-data routing phase. In the sequence of the thesis, we detail the procedures of routers and the controller.

2.1 Bootstrap Phase

In the Bootstrap phase, routers find the controller to register themselves, the controller acquires information to construct the global topology, and the controller calculates all routes. After this phase, the controller can install the routes, forwarding rules, on routers because it knows all routes to any router in the network. Figure 2.3 shows the three procedures of Bootstrap phase: the Hello, the Controller Discovery, and Router Registration. These are essential procedures for the net-

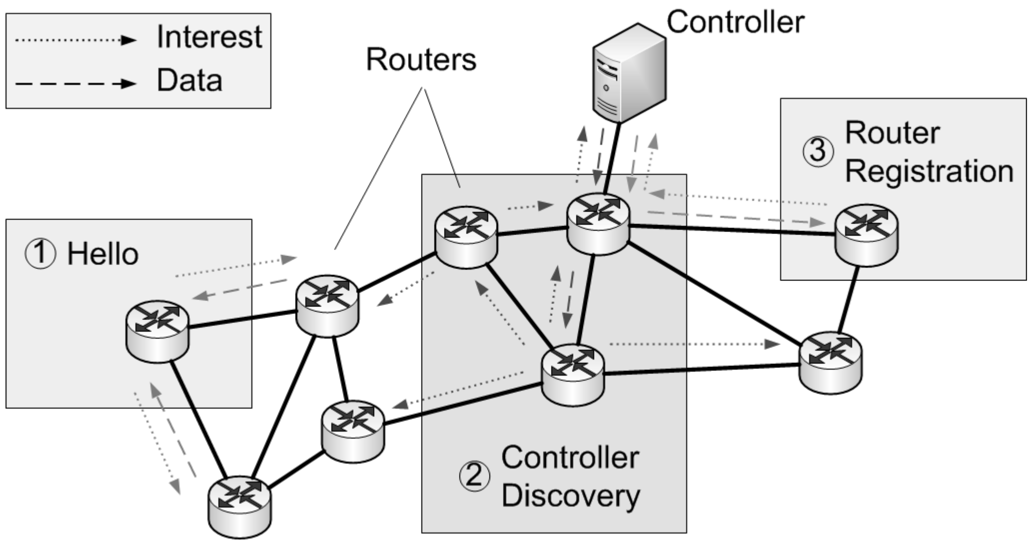


Figure 2.3: The three procedures of CRoS-NDN Bootstrap phase: 1) Hello, in which a router announces its presence to neighbors; 2) Controller Discovery, when routers flood interests to discover the controller location; and 3) Router Registration, when routers send their neighborhood to the centralized controller that assembles the information pieces and constructs the global network topology.

work Bootstrap phase, but CRoS-NDN router periodically runs the Hello procedure to monitor the connectivity to neighbors, and register any connectivity change in the controller to maintain the routing information updated. Next, we describe the procedures and algorithms to discover routes.

2.1.1 Hello

All routers send a Hello interest packet to inform their directly connected neighbors about their presence. Figure 2.3 item 1 shows a router sending Hello interests to its neighbors and Figure 2.4 presents the behavior diagram in Specification and Description Language (SDL). In Hello, routers diffuse periodic interest packets with name `/hello/routerID`[3] on all interfaces. Every router that receives an interest packet with prefix `/hello` replies with a data packet. The first reply cleans the respective PIT entry, but the Hello data packet contains no routing relevant content and the router discards subsequent data packets replies.

Routers install a FIB entry to prefix `/router/routerID` via the incoming interface of the `/hello/routerID` interest. Routers also store the received routerID in their local neighbor list and routers periodically remove recently unheard routerIDs from neighbor list. Therefore, each router keeps locally a restricted view of the

[3]We omit the sequence numbers in content names for simplicity, `/hello/routerID/sequence1` for example, but different sequence numbers refer to different contents avoiding responses from local cache.

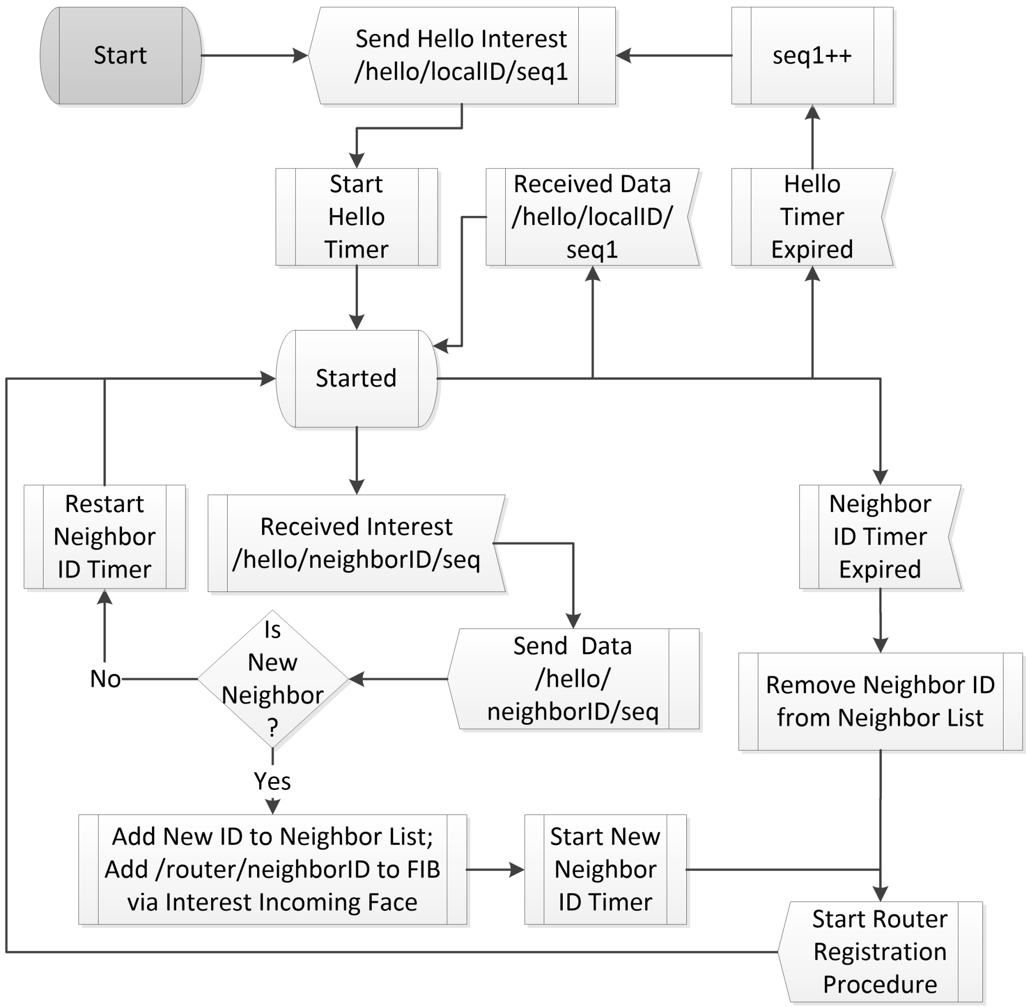


Figure 2.4: SDL behavior diagram for routers execution of Hello procedure.

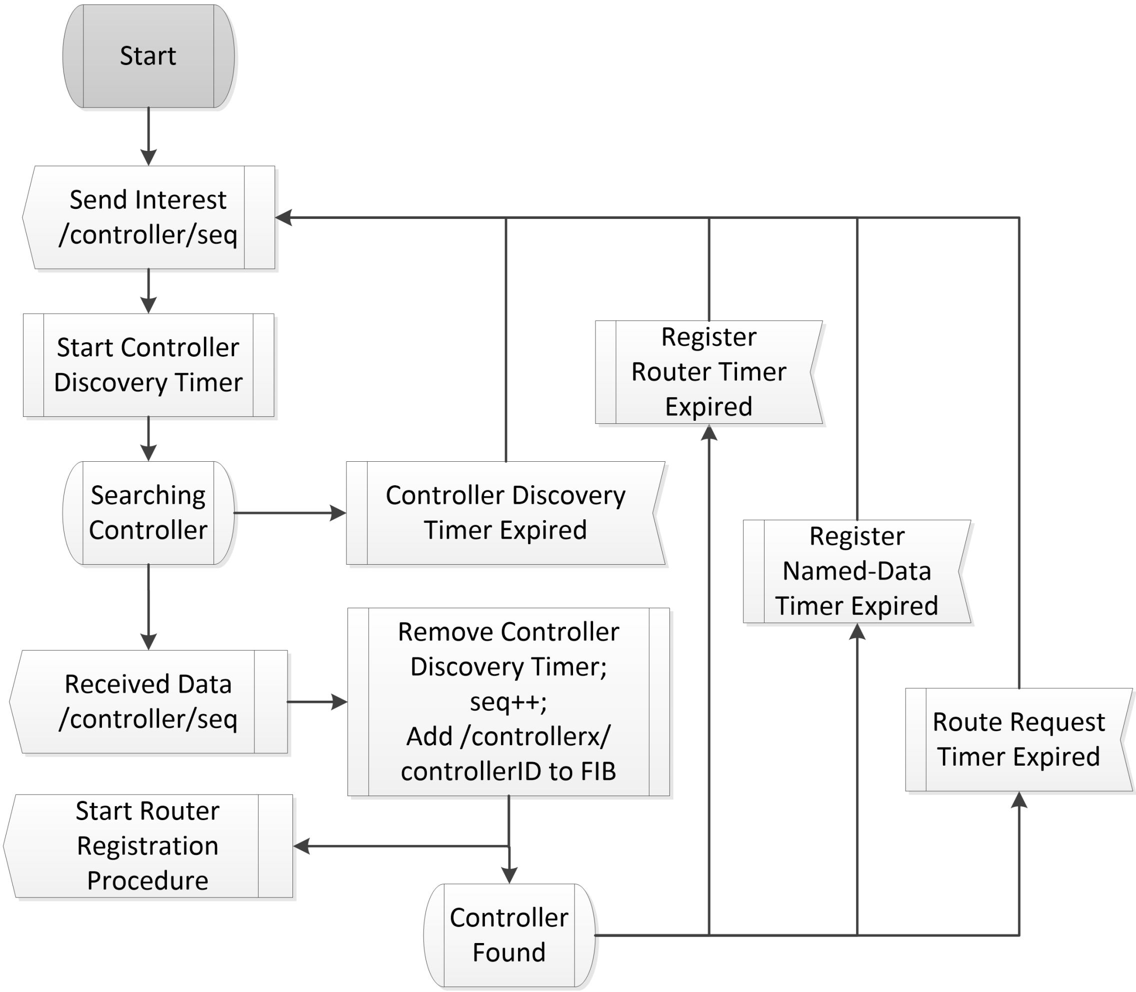


Figure 2.5: SDL behavior diagram for routers execution of Controller Discovery procedure.

network topology. The rate of Hello interests define the timer to remove unheard neighbors and all routers employ an equal rate.

2.1.2 Controller Discovery

Routers initially do not know where the controller is located and, thus, routers asynchronously flood, on all interfaces, interest packets with name `/controller` to discover routes to the controller. When a router receives the `/controller` prefix interest packet, it adds to its PIT and forwards the interest packet to all interfaces, except the incoming interface. When the controller receives the interest, it replies with a data packet containing its ID. The router that receives the data packet stores it in its CS and the router forwards the data packet to the downstream path. Figure 2.3 item 2 illustrates the flooding of controller discovery interests. The router also installs a FIB entry to prefix `/controllerx/controllerID` via the data-packet incoming interface and stores the controller ID in a controllers table.

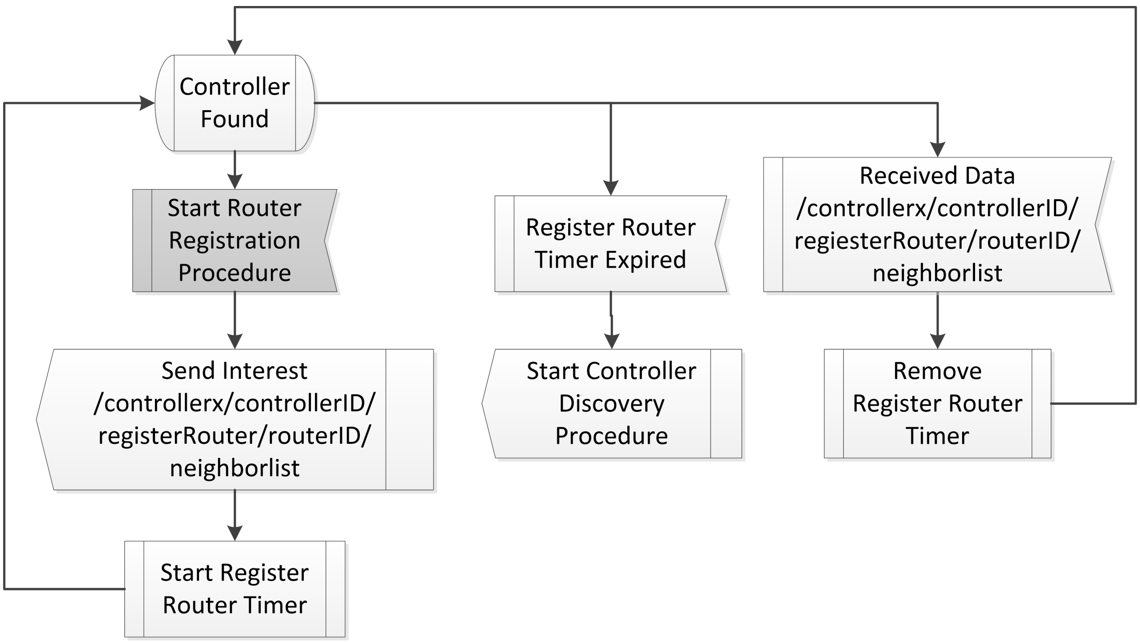


Figure 2.6: SDL behavior diagram for routers execution of Router Registration procedure.

All routers receive the data packet because they sent or forwarded the interest packet and, thus, all routers know a path to the controller. Routers cache the data packet to reply subsequent interest. Caching controller data reduces the interest flooding through local responses. Additionally, each router restarts the Controller Discovery only upon time expiration of no response interests to controller. Figure 2.5 presents the SDL behavior diagram for the Controller Discovery procedure executed by routers. It is worth to note that Hello and Controller Discovery procedures start simultaneously.

2.1.3 Router Registration

When the router finds the controller, the router registers itself in the controller, as depicted in Figure 2.3 item 3. The router sends an interest packet with name `/controllerx/controllerID/registerRouter/routerID/-neighbor1/.../neighborN`, where `controllerID` is the known controller ID, `registerRouter` indicates the request for registration of the `routerID`, and the sequence of identifiers `neighbor1/.../neighborN` is the router neighbor list. When the controller receives this interest packet, the controller creates or updates a router entry in its internal routers table adding the router neighbor list. Additionally, the controller replies the interest with an acknowledgement data packet with no routing relevant content.

Whenever a router detects a topology change, this router restarts a new Router

Registration procedure in order to guarantee an updated global topology at the controller. Nevertheless, whenever a router does not receive a response for the registration request, the router restarts the Controller Discovery procedure. Figure 2.6 presents the SDL behavior diagram for the Register Router procedure executed by routers.

The Route Calculus Algorithm

After the Router Registration procedure, the controller knows all routers in the network and their respective neighbors list. Therefore, the controller constructs the network topology and calculates the routes between any two routers. The controller recalculates the routes whenever it perceives any topology change. The controller can also calculate alternative routes with higher costs to distribute the bulk traffic in the network and improve overall performance [58]. This scheme may be extended to inform the controller about the bandwidth and latency on each link and improve the path costs in route calculation, but it is out of the scope of this thesis.

2.2 Named-Data Routing Phase

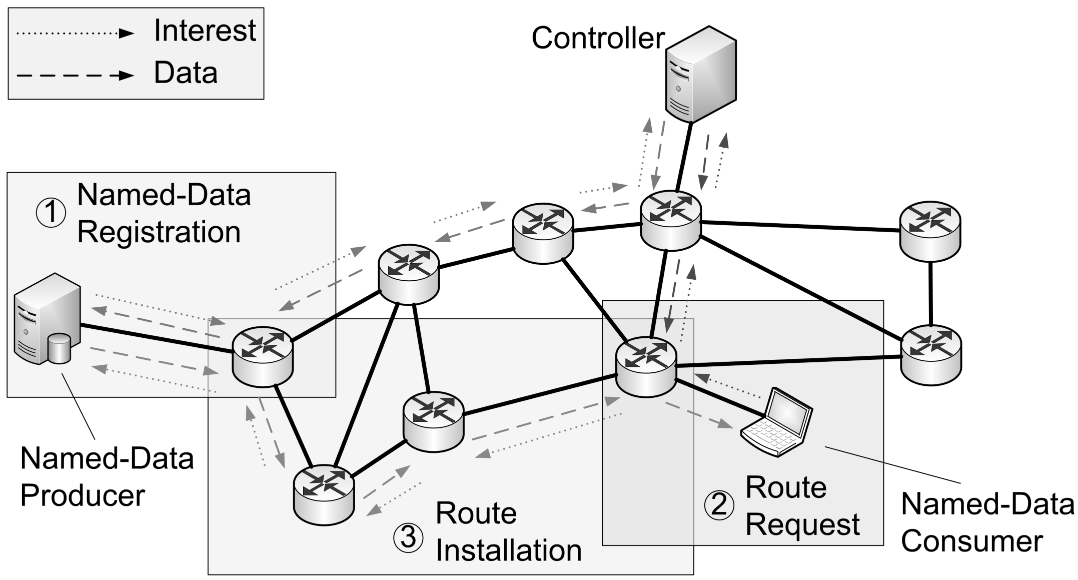


Figure 2.7: The three procedures of CRoS-NDN Named-Data Routing phase: 1) Named-Data Registration, when producers register new named-data in the controller; 2) Route Request, when routers ask the controller for routes to unknown prefixes; an 3) Route Installation, when routers install the requested route to the producer.

After the Bootstrap phase, all routers can send messages to the controller, the controller knows the global network topology, and the controller has already calculated routes from one router to the others. However, the controller does not know the location of named-data. Therefore, producers must register the named-data in the controller, and routers can install routes to named-data based on specific interest packets as depicted in Figure 2.7.

2.2.1 Named-Data Registration

When the producer publishes a new unregistered named-data or a new content copy location, it sends an interest packet with name `/registerNamedData/myprefix`, where `registerNamedData` indicates the `myprefix` registration intention. When the connected router receives this packet, it adds a PIT entry and also adds a FIB entry to `myprefix` via the incoming interface. Following, instead of forwarding the interest packet to one of its interfaces, the router generates a new interest packet with name `/controllerx/controllerID/registerNamedData/routerID/myprefix` to indicate the named-data location and sends it directly to the controller. When the controller receives this interest it stores `routerID` as the location of `/myprefix` in named-data location table. If the controller already received a registration request from part of that prefix at the same location, it can optionally aggregate the prefixes in a single entry. The controller replies a data packet to the router, which replies a data packet to the producer and acknowledges the registration. This procedure restricts to controller the consolidation of data location/mobility and, thus, it reduces the control message overhead. Figure 2.7 item 1 illustrates the message flow for a producer requesting the registration of named-data. Figure 2.8 presents the SDL behavior diagram for the Register Named-Data procedure executed by routers upon producer request.

It is worth to note that the decision about the registration of content copies is an open issue and this issue is outside the scope of our thesis. Therefore, we do not consider the registration of content copies stored in router caches. However, we do consider that consumers can store content copies locally and register their location based on local policies.

2.2.2 Route Request and Route Installation

When any node requires a named-data, it sends an interest packet with name `/wantedcontent`. The first router receives this packet and adds to its PIT. If the router FIB contains no forwarding rule, the router should discover the route. Then, the router generates a route-request interest with name `/controllerx/controllerID/routeFrom/sourceRouterID/wantedcontent` [4], where `routeFrom/source-`

[4]The router requests the route only if it is the first hop, there is no pending route-request interest, and no pending route-install interest for the prefix of the `/wantedcontent` name. The

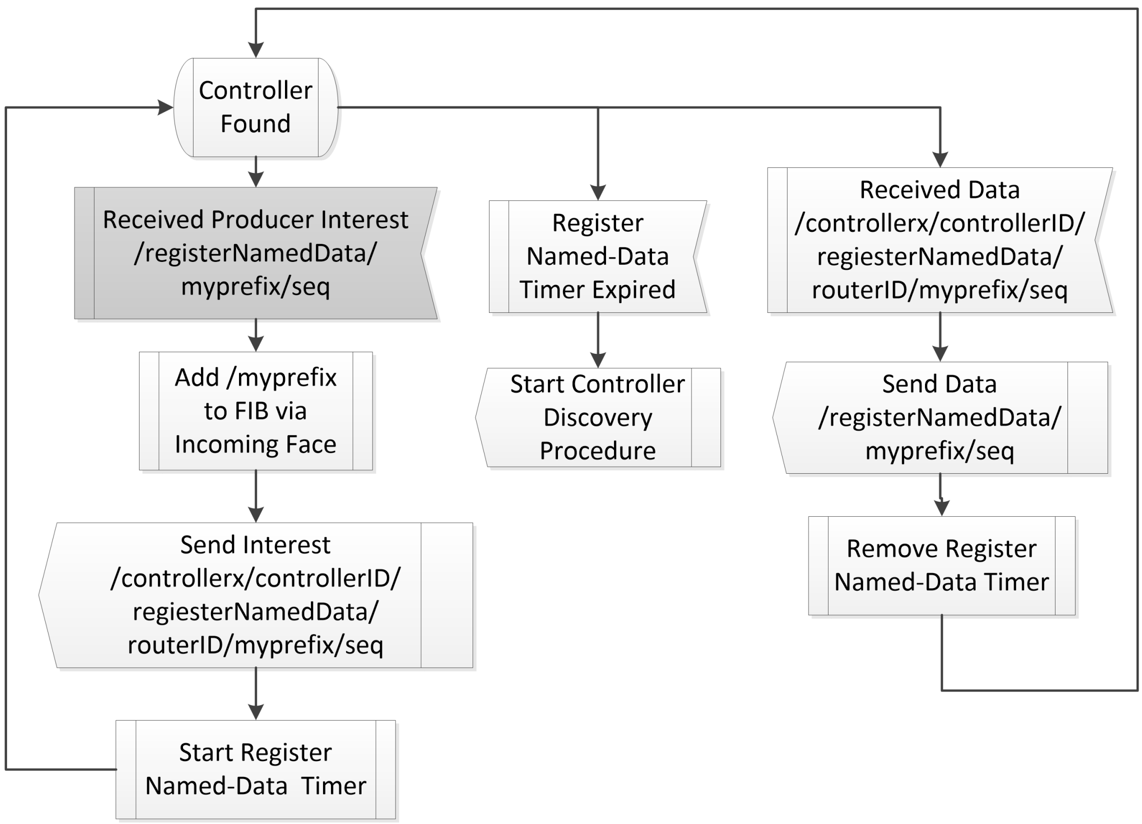


Figure 2.8: SDL behavior diagram for routers execution of Named-Data Registration procedure.

`RouterID` indicates the source router of the Route Request interest. The `prefix` for the `wantedcontent` is already registered and, therefore, the controller knows the route destination. The controller searches the registered copies of the requested content, evaluates the copy with the lowest route cost from source to destination, and generates the route-reply data packet informing the registered prefix for the content.

The route-reply data packet also contains the whole route from the source router to the destination, which includes the information of the route-installation interest name to be generated by the source router. The controller may also include alternative routes in the route-reply data packet.

When the source router receives the route-reply data packet, the router searches the next hop information in the packet content. Next, the router adds a FIB entry to the `/prefix` via the same interface it reaches the next hop router.

In the sequence, the source router creates a route-install interest packet with name `/router/hopID1/installRouteAndForward/hopID1/.../hopIDn/prefixSize/wantedcontent`, where `hopID1/.../hopIDn` is the routers sequence in the path to the prefix producer and, the `prefixSize` indicates the size of content prefix registered by the producer. The next hop router replies the interest packet and employs the sequence to create the FIB entry to the `/prefix`. This procedure is repeated until the interest packet reaches the destination router, which already contains a FIB entry to `/prefix` and the route is fully installed. Each router in the path adds a PIT entry to `/wantedcontent` and, the last hop router sends an interest to `/wantedcontent` that is replied back until the consumer. Therefore, one single interest installs the new route and requests the content. Figure 2.7 item 3 illustrates the Route Installation message flow. Figure 2.9 shows the SDL behavior diagram for the Route Request and Route Installation procedures executed by routers.

Whenever a PIT entry lifetime expires, the router removes the FIB entry employed for interest forwarding [5]. The subsequent interest for the name prefix does not match any FIB entry and, then the router requests a new route to the controller restarting the Route Installation procedure. The procedure reactively updates the

---

first hop checking assures that only access routers do request routes. The pending interest checking avoids redundant route request for the same prefix. If it is not the first hop router, the router replies back an Interest Nack to indicate failure to find content [6]. We omitted the Nack in the sequence diagram of Figure 2.2 for simplicity. We note that there is an ongoing discussion about Nacks implementation [59] and we consider Nacks are an improvement to our scheme that accelerates route update, but Nacks are not an essential component and can be implemented as Interest or Data packets.

[5] We consider the lifetime is higher than and close to the maximum round trip delay in the network. We note that the higher is the PIT entry lifetime, the higher is the delay to remove the respective FIB entry. In addition, the higher is the lifetime, the higher is the PIT memory requirement [60–62]. On the other hand, a lifetime lower than the round trip delay causes interest retransmission and may cause interest loop [63]. The mechanism to autonomously estimate the lifetime is not covered in this thesis.

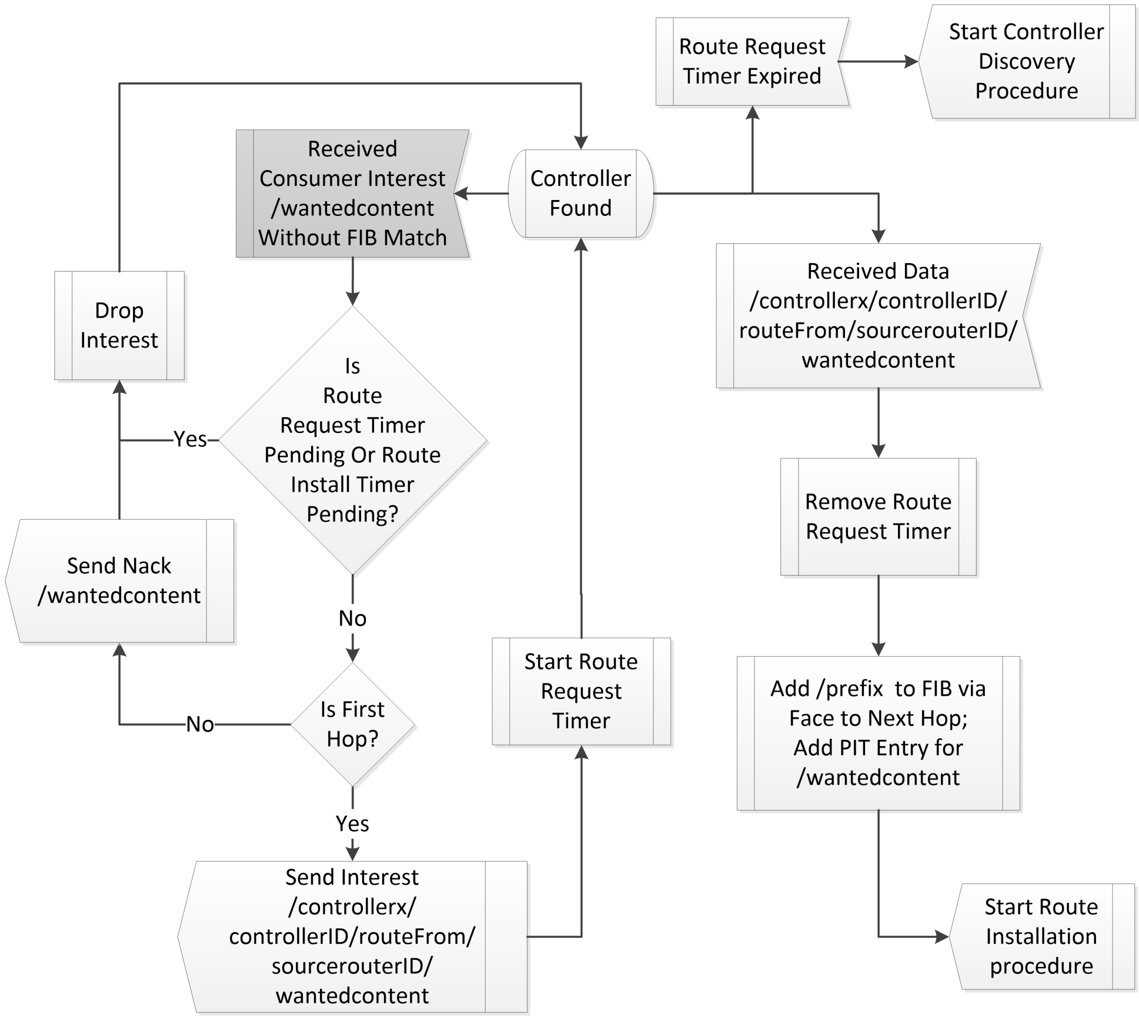


Figure 2.9: SDL behavior diagram for routers execution of Route Request procedure.

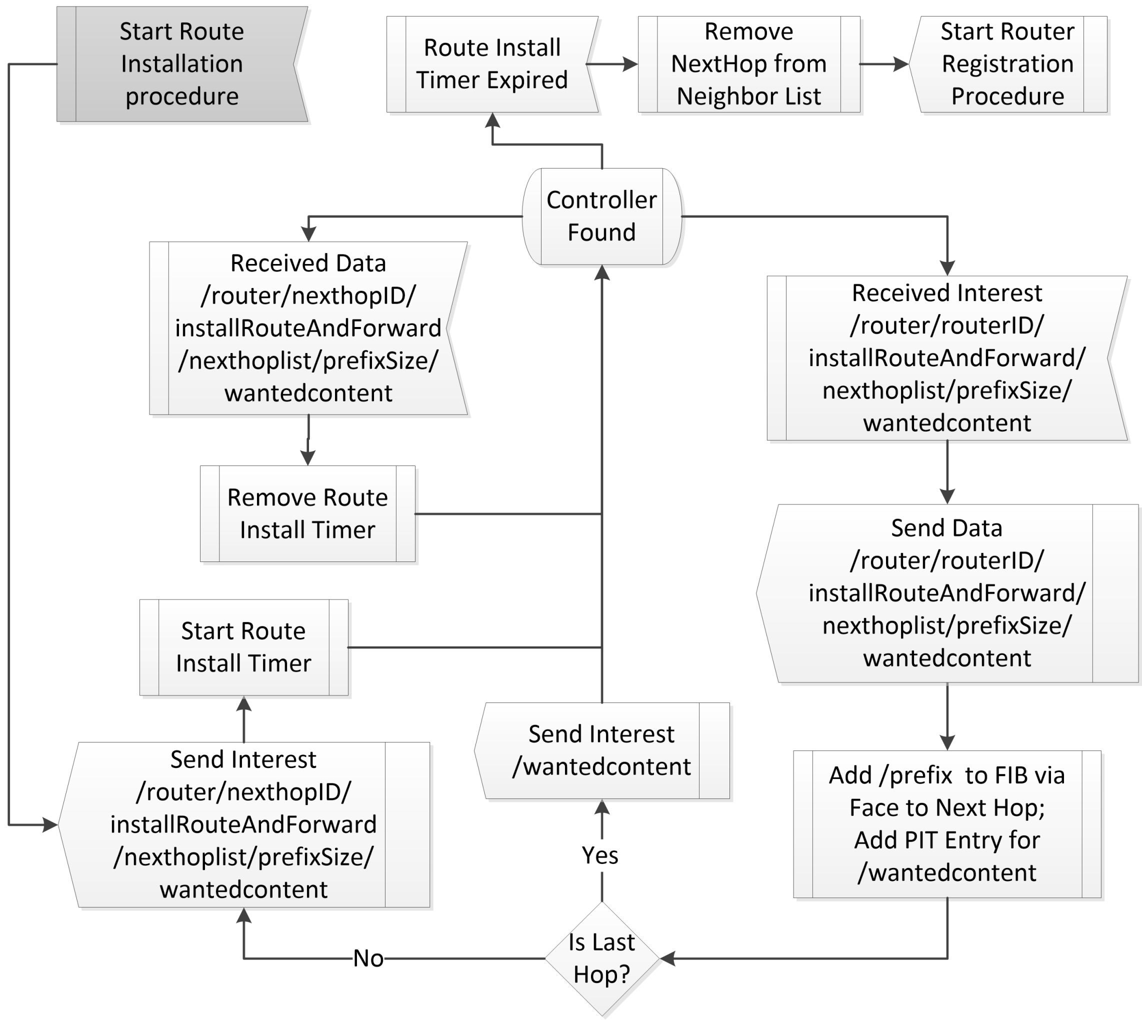


Figure 2.10: SDL behavior diagram for routers execution of Route Installation procedure.

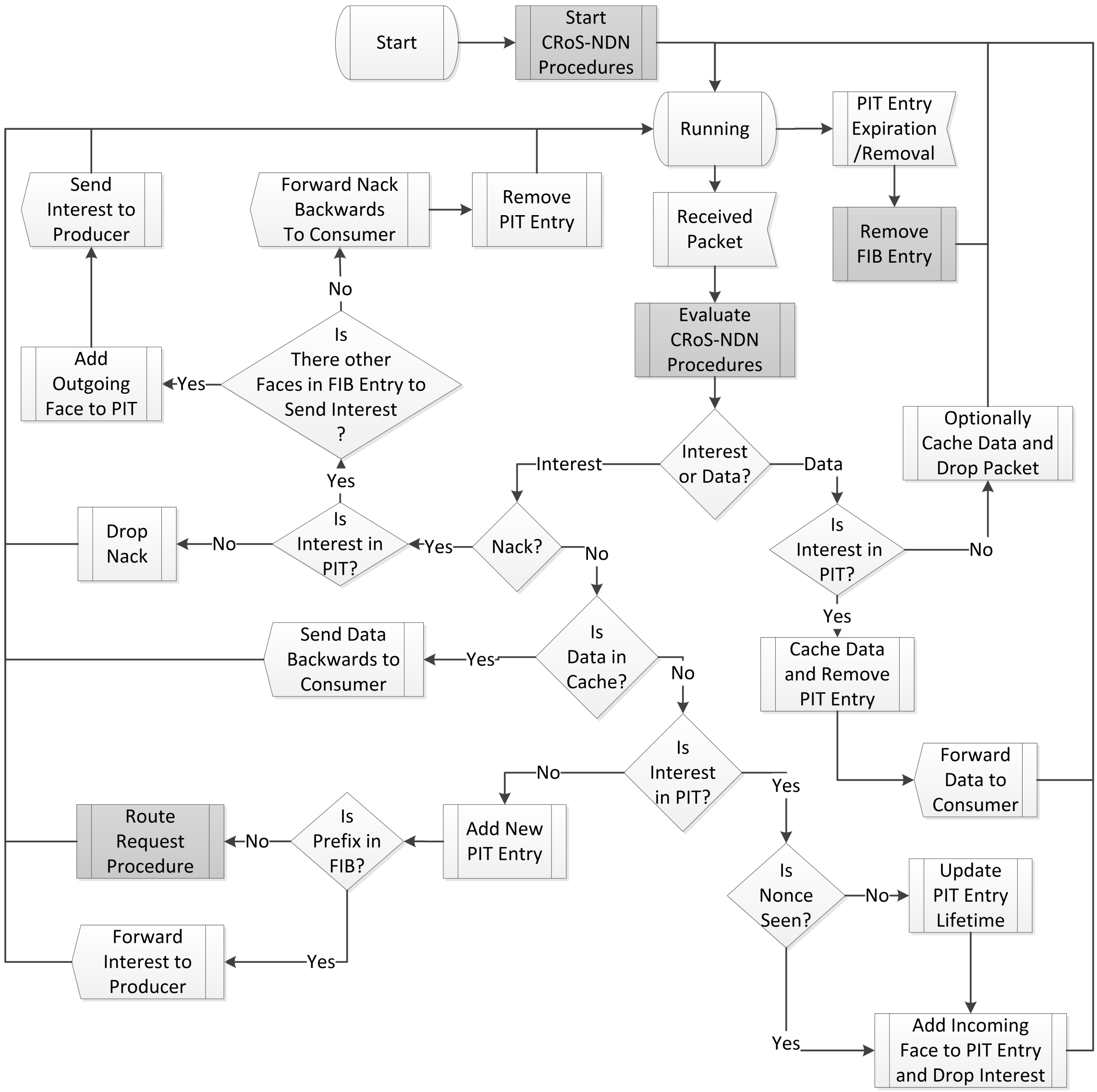


Figure 2.11: SDL behavior diagram for CRoS-NDN routers.

router data plane with the controller network view, providing an efficient approach for routing coordination between routers and controller. In special, if a route-install interest expires without response, the source router removes the next hop router from the neighbor list and, next, it updates the controller before the next Hello procedure execution. Therefore, the Route Installation procedure triggers the control plane update for topology changes in paths to requested content, and, thus it reduces the convergence delay irrespective the Hello interest rate. Furthermore, this procedure updates invalid routes on-demand upon data mobility and it avoids the proactive replications of routing information from controller to all routers.

Figure 2.11 shows the SDL behavior diagram for CRoS-NDN routers. CRoS-NDN procedures start and packets follow the default NDN protocol. Whenever a packet arrives, the CRoS-NDN router evaluates the packet according to Hello, Controller Discovery, Router Registration, Named-Data Registration, Route Request, and Route Installation procedures. In the sequence, NDN routers identify received packets as Interest or Data. In case of Interest, the NDN router identifies the packet as Nack or normal Interest. In case of Nack, the NDN router drops Nacks that have no corresponding PIT entries. In case of PIT entry match, the NDN router checks alternative faces to send the normal Interest in direction to content. Otherwise, the NDN router forwards the Nack backwards in direction to consumer following the incoming faces that are registered in PIT entry and, then, the NDN router removes the PIT entry. In case of normal Interest, the NDN router checks cached data and, in case of cache hit, it sends back to consumer the Data packet. Otherwise, the NDN router checks if there is a PIT entry corresponding to this Interest and, in case of match, the NDN router identifies if this Interest has a known nonce. In positive case, the NDN router adds the incoming face to PIT entry and drops the Interest. Otherwise, the NDN router updates the PIT entry lifetime, adds the incoming face to the PIT entry, and drops the Interest. In case of Interest that is not in PIT, the NDN router adds a new PIT entry and checks if there is a matching FIB entry. In positive case, the NDN router forwards the Interest to producer. Otherwise, the CRoS-NDN router starts the Route Request procedure. In case of Data, the NDN router caches and forwards data packets to faces indicated in PIT entry. When there is no corresponding PIT entry, then the NDN router optionally caches the data and discards the packet.

Chapter 3

Distributed Routing Schemes for Named-Data Network

In this section, we review the main distributed routing schemes for Named-Data Network that are cited in the literature. Two schemes are based on IP counterparts and the other one is a specific scheme for NDN. We use the "Like" term to denote our own implementation for each scheme due to the unavailability of the source code. It is of utmost importance to note, first, that our implementation reflects the main limitations of each scheme concerning the messages exchange and, second, that known optimizations that we found in the literature for the IP counterparts do not overcome these limitations.

3.1 Address Resolution Protocol Like

The Address Resolution Protocol Like (ARPLike) routing scheme, based on proposals [64–66] and IP counterpart [67], employs a consumer-oriented approach to find content. ARPLike reacts to consumer requests flooding the network with interests for content that have unknown forwarding rules. Each router floods the network whenever the incoming interest does not match any FIB entry. Upon content response arrival, ARPLike router updates its FIB adding a new entry with the content name prefix pointing to the content incoming interface. Routers directly forward the subsequent interests with the same prefix using the new FIB entry. ARPLike employs the same CRoS-NDN procedure to remove invalid FIB entries, i.e., a PIT entry expiration timeout triggers the removal of the associated FIB entry. Figure 3.1 presents the interest time sequence for ARPLike procedures.

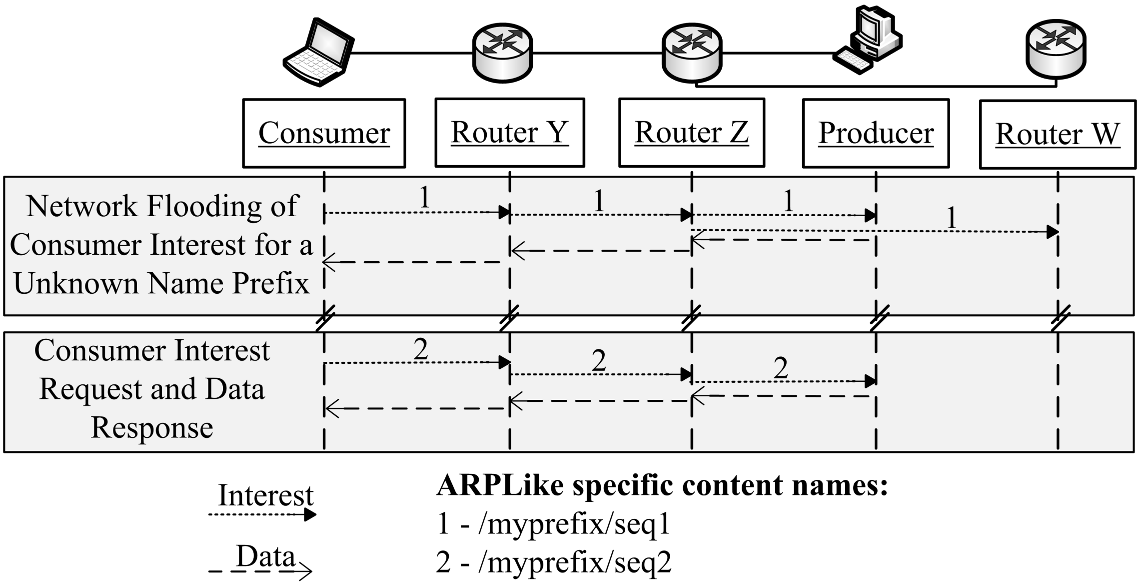


Figure 3.1: The Interest/Data packet sequence for ARPLike scheme procedures. (1) Consumer requests the content and Routers Y, Z, and W flood the request on all interfaces looking for content. (2) Routers directly forward further interests to the originating interface of data response for the first interest.

3.2 Open Shortest Path First Like

The Open Shortest Path First Like (OSPFLike) routing scheme follows CCN original routing concept [4]. OSPFLike employs a producer-oriented approach to announce content availability in a pro-active fashion. Unlike CRoS-NDN, OSPFLike content-producers periodically flood the entire network with prefix announcing interests. Each router does not monitor the connectivity to its neighbors and, therefore, routers forward the prefix announcement interest to periodically update the path to producer. Network wide recurrent flooding increases the routing signaling overhead in proportion to network size and to the number of content prefixes.

In order to flood the network, producers add a special prefix to content announcement interest messages. This prefix triggers two actions on interest-receiving routers: i) the router diffuses (replicates) the interest to all its interfaces and ii) the router adds a new FIB entry with the announced prefix pointing to the announcement-incoming interface. OSPFLike employs the same CRoS-NDN procedure to remove the invalid FIB entries, i.e., PIT entry expiration timeout triggers the removal of the associated FIB entry. Furthermore, unlike the IP Routing Information Protocol (RIP) [68], NDN PIT entries prevent loops in OSPFLike prefix announcements.

Unlike the IP conterpart [69], OSPFLike routers do not maintain a link-state database view of the topology. In addition, unlike our CRoS-NDN scheme, OSPFLike routers have no knowledge of network topology. Therefore, OSPFLike forwarding decisions follow the local view of the received prefix announcements. If a router receives the same announcement from multiple interfaces, then, it ranks

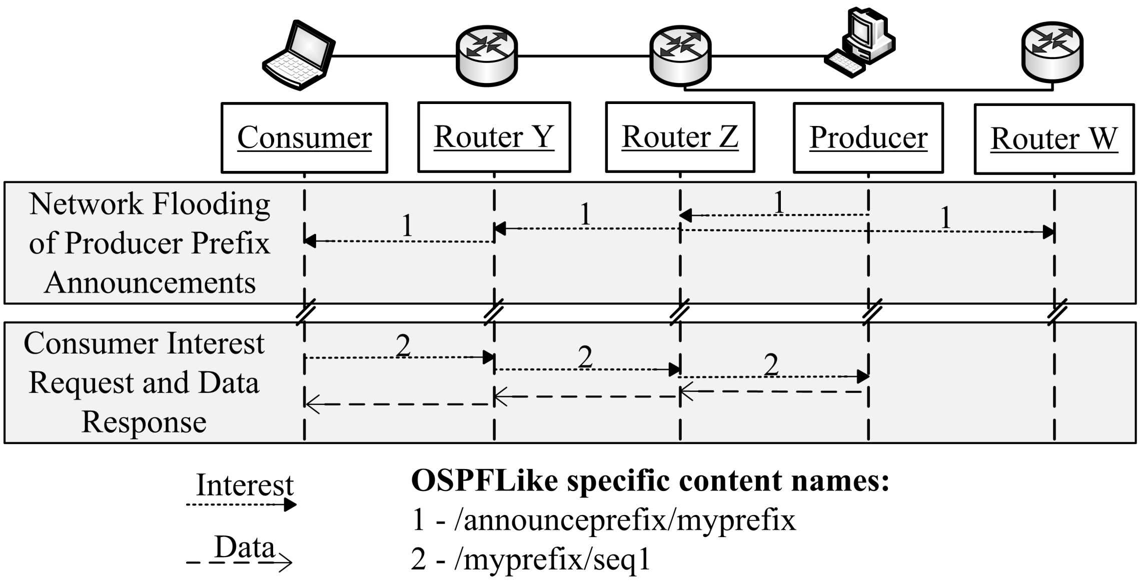


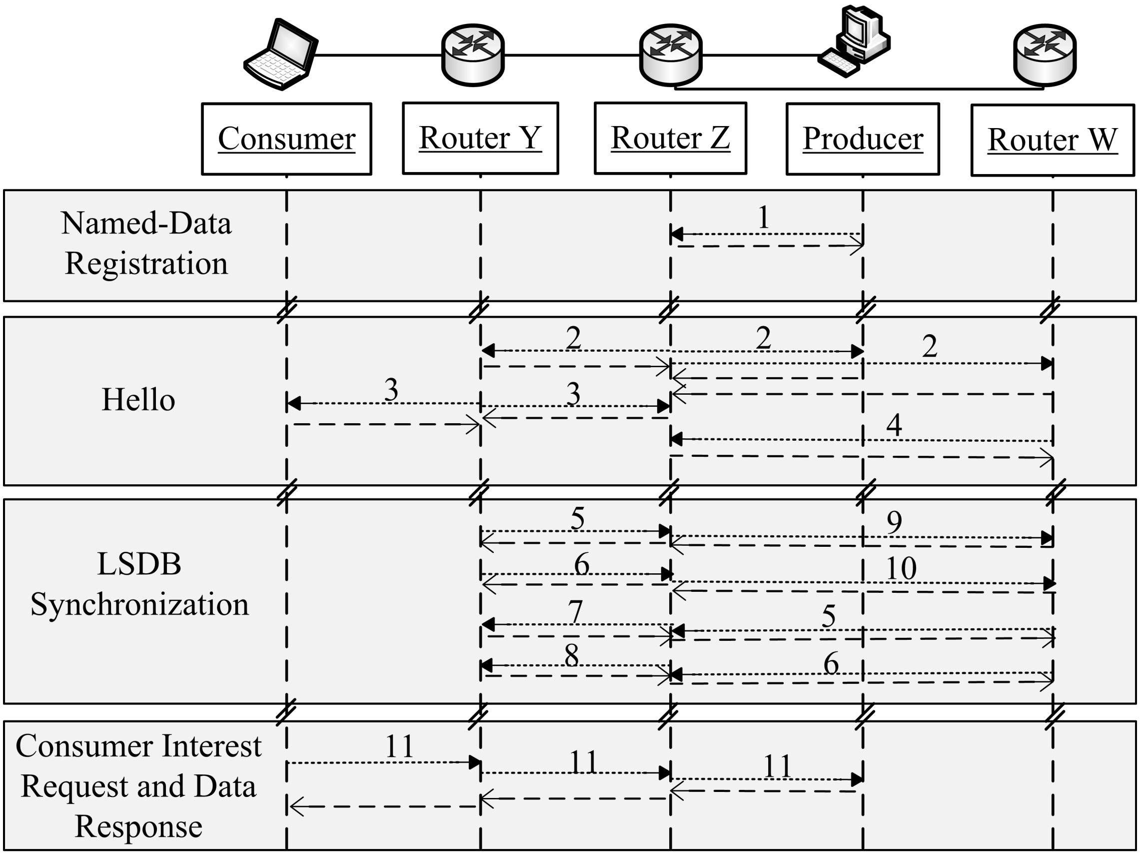
Figure 3.2: The Interest/Data packet sequence for OSPFLike scheme procedures. (1) Producer announces the prefix of available named-data. Afterwards, each router installs a FIB entry for the prefix and floods the prefix to its other interfaces. (2) Consumer requests and receives the content.

output interfaces according to hop distance to producer. Moreover, unlike CRoS-NDN, OSPFLike router stores all available content prefixes simultaneously on its FIB memory. Figure 3.2 presents the interest time sequence OSPFLike procedures.

3.3 Named-Data Link State Routing Like

The Named-Data Link State Routing Like (NLSRLike) routing scheme, based on proposals [70, 71], avoids the OSPFLike flooding procedure. It replaces the OSPFLike periodic flooding of prefix announcements by a hop-by-hop procedure for database synchronization. Unlike the preceding schemes, each NLSRLike router maintains the full view of the network in a local database called Link State DataBase (LSDB). The LSDB stores the network topology and the content producer locations using database entries called Link State Advertisements (LSAs). The neighbor-LSA, with name `/routerid/LSAtype1/version`, stores the router adjacency list and the prefix-LSA, with name `/routerid/LSAtype2/LSAid/version`, stores the association of the content prefix with the producer identifier. Each router computes the hash for each LSA name, builds a tree with branches based on LSA name prefixes, and sums the hashes of LSA names that share equal prefix to compute the hash for each branch. NLSRLike builds a hash tree for the prefixes of LSA names and the LSDB hash is the root hash of this tree.

Producer registers the content prefix in its access router, Named-Data Regis-



NLSRLike specific content names:

1 - /registerNamedData/myprefix
2 - /hello/RouterZ/hashZ
3 - /hello/RouterY/hashY
4 - /hello/RouterW/hashW
5 - /RouterZ/gethashes/size/prefix/hash
6 - /RouterZ/getlsa/size/lsaname
7 - /RouterY/gethashes/size/prefix/hash
8 - /RouterY/getlsa/size/lsaname
9 - /RouterW/gethashes/size/prefix/hash
10 - /RouterW/getlsa/size/lsaname
11 - /myprefix/seq1

Figure 3.3: The Interest/Data packet sequence for NLSRLike scheme procedures. (1) Producer announces the content prefix to Router Z. (2, 3, and 4) Routers Z, Y, and W periodically announce their presence and the hash of their local database. (5, 6, 7, 8, 9, and 10) Routers Z, Y, and W synchronize theirs database. (11) Consumer requests and receives the content.

tration procedure. Then, the router updates its local LSDB with a prefix-LSA. Neighbor routers exchange periodic interests to identify router adjacency, verify local connectivity, and compare their LSDB hashes (Hello procedure). Each router registers its neighbors in its local LSDB with a neighbor-LSA. If LSDB hashes of two neighbor routers differ, these routers initiates the LSDB Synchronization procedure that recursively exchange the branch hashes of LSA name prefix with hash differences until the branch leaf is reached. Then, the LSA difference is updated. Each router builds the network topology and the content prefix to producer identifier map based on its local LSDB and, then, the router evaluates locally the output interface upon consumer interest reception. Figure 3.3 presents the interest sequence for NLSRLike procedures.

Chapter 4

Performance Analysis

In the preceding section, we have presented different routing schemes for content location and forwarding. In the ARPLike scheme, the consumer floods the network with interest packets to obtain a content. In other words, it is a consumer-oriented reactive flooding procedure. On the other hand, OSPFLike and NLSRLike proactively announce content localization, routing information, on the network, being a producer-oriented approach. OSPFLike periodically floods the network with announcements while NLSRLike employs a hop-by-hop procedure. Unlike the preceding schemes, our CRoS-NDN proposal avoids network recurrent flooding by consolidating network information on the controller. In this section, we analyze the Data Delivery Efficiency (DDE) and the Data Delivery Delay (DDD) for each routing scheme. The data delivery delay measures the delay between consumer content request and consumer content reception. The data delivery efficiency measures the ratio of the consumer-received data packets and the number of interest packets sent on each network link [1]. Therefore, local cached data on consumers yields delay zero and efficiency one. We derive mathematical expressions for DDE lower bound and DDD upper bound, worst case scenario. We employed the obtained expressions to discuss the limitation factors of each scheme. Table 4.1 presents the considered parameters employed for deriving the routing scheme expressions. The analysis considers that multiple collocated nodes ran the CRoS-NDN controller function as a single entity and that these nodes share a database that stores both the named-data location and the routers adjacency information for a single domain. This assumption does not invalidate the comparative performance analysis because it relies on data center infrastructure to host the nodes and, therefore, it eliminates processing

[1]The data delivery efficiency is a footprint reduction metric [72]. This metric indicates *how many* cache hits and *where* they happen along a path to content on average in the network. To the best of our knowledge, this metric is the best choice to jointly evaluate the routing-scheme message overhead and the cache hits. We note that coupling factor metrics, which indicate cache hits concentration close to the edge or the core of the network [72], are more suited to content placement policies [12, 73, 74], but the evaluation of these policies is not covered in this thesis.

Table 4.1: Parameters of the routing scheme expressions.

Parameter - Description	Parameter - Description
N - Number of **N**odes	TR - **T**opology change **R**ate
L - Number of **L**inks	LD - **L**ink **D**elay
H - Network diameter **H**ops	DDE - **D**ata **D**elivery **E**fficiency
CR - **C**onsumer **R**ate	RTD - Max **R**ound **T**rip **D**elay
AP - **A**nnounced **P**refixes	CD - **C**onsumer-producer **D**elay
AR - **A**nnouncement **R**ate	AD - **A**nnouncement **D**elay
FF - **F**IB match **F**ail ratio	TD - **T**opology-update **D**elay
KR - **K**eepalive **R**ate	DDD - **D**ata **D**elivery **D**elay

Table 4.2: Data delivery efficiency lower bound expressions.

Scheme	Data Delivery Efficiency (DDE)
ARPLike	$1/(FF.L+(1-FF)H)$
OSPFLike	$CR(1-FF)/(AP.L.KR+CR.H)$
NLSRLike	$CR/(L(2.KR+4.TR+5.AR)+CR.H(FF+1))$
CRoS-NDN	$CR/(L(2.KR+TR)+H(N.TR+AR+CR(FF+1)))$

power and storage bottlenecks of a single node.

Table 4.2 presents the lower bound values for data delivery efficiency of each routing scheme. The expressions consider that all network links have the same **L**ink **D**elay, LD, each **C**onsumer sends interests and receives data with a constant **R**ate, CR, consumer to producer distance equals network diameter [2] **H**ops, H (worst case scenario), and router to controller distance equals network diameter (worst case scenario). It is worth to note that, in this scenario without cache, the lower-bound for the optimum efficiency equals $1/H$. In the sequence, we detail the expressions for each scheme.

ARPLike efficiency depends on the fraction of interests that match an existing FIB entry, FIB rule hit ratio, which is equal to the complimentary probability of FIB match **F**ail **F**raction, $1 - FF$, FIB rule miss ratio. ARPLike router straightly forwards to producer interests with FIB match. If an interest does not have a FIB match, the router floods the interest in its links. The higher is the fraction of directly forwarded interests, $1-FF$, the closer ARPLike efficiency becomes to the optimum value that the lower bound is equal to $1/H$. The higher is the fraction of flooded interests, FF, the lower is the ARPLike efficiency. In large networks with restricted diameter ($L >> H$), if consumer traffic shows uncorrelated interest prefixes and router FIB has insufficient memory to support all content prefixes simultaneously,

[2]We consider the diameter equals the maximum value of all the shortest paths between any combination of router pairs.

then, ARPLike router recurrently floods interests and the efficiency tends to zero due to FIB entry replacement. Under router unbounded FIB memory assumption and after enough time, ARPLike routers store routes to all prefixes and FIB match failure tends to zero, and in this case, ARPLike efficiency tends to the optimum value. Inequation 4.1 shows the lower bound expression for ARPLike data delivery efficiency.

$$DDE_{ARPLike} >= \frac{1}{(FF.L + (1 - FF)H)} \tag{4.1}$$

In OSPFLike scheme, the number of interests on the network depends on the rate of consumer interests, CR, the rate of periodic content announcements, KR[3], and the number of **A**nnounced **P**refixes, AP. Consumer interests traverse H links to reach producer, expressed by $CR.H$ denominator element. OSPFLike periodically floods all announced prefixes, AP, on all network links, L, with rate KR, given by $AP.L.KR$ denominator element. The number of content data received by the Consumer is equal to the fraction of consumer interest rate that match a FIB entry and, thus $CR(1 - FF)$ is the numerator of the efficient expression. OSPFLike efficiency decreases with the number of content prefixes, AP, the rate of periodic prefix announcements, KR, and the number of networks links, L. Inequation 4.2 shows the lower bound expression for OSPFLike data delivery efficiency.

$$DDE_{OSPFLike} >= \frac{CR(1 - FF)}{(AP.L.KR + CR.H)} \tag{4.2}$$

NLSRLike routers monitor their neighbors sending keep alive interests on all links, by Hello procedure, corresponding to $2.L.KR$ messages in efficiency denominator. Additionally, the LSDB Synchronization procedure of NLSRLike takes, respectively, five and four interests per link to synchronize new prefix-LSAs and router adjacency LSAs, given by $L(5.AR + 4.TR)$ denominator element[4]. Producers announce new prefixes with rate AR and topology changes with rate TR. Furthermore, besides the consumer to producer interest hops given by $CR.H$, NLSRLike FIB match failure FF takes one interest to control plane per router in the path from consumer to producer expressed by $CR.H.FF$. Inequation 4.3 shows the lower bound expression for NLSRLike data delivery efficiency.

$$DDE_{NLSRLike} >= \frac{CR}{(L(2.KR + 4.TR + 5.AR) + CR.H(FF + 1))} \tag{4.3}$$

The numerator of the lower bound expression for CRoS-NDN efficiency corre-

[3]This value corresponds to the **K**eep alive **R**ate, KR

[4]Prefix-LSA and router adjacency LSA names have 4 and 3 components, respectively. Thus, the LSDB synchronization takes 4 and 3 interests to navigate from the root to the leaf of the LSDB hash tree and one additional interest to update the new LSA.

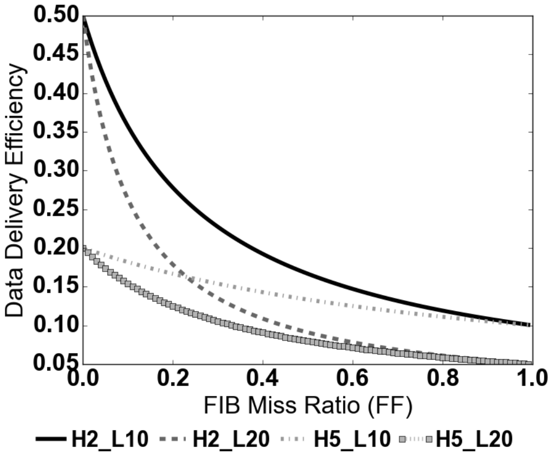
sponds to the consumer received content rate and it equals the consumer interest request rate, CR. The denominator is composed of: $2.L.KR$ element that corresponds to the Hello procedure of router interests monitoring its neighbors; $TR.L$ element corresponds to Controller Discovery procedure, when controller discovery interests are flooded after each topology change; $H.N.TR$ corresponds to the Router Registration procedure, when all routers register in controller after each topology change; $H.AR$ corresponds to the Named-Data Registration procedure, when producers register available content prefixes on controller with rate AR; $H.CR.FF$ corresponds to the Route Request procedure, when consumer sends to controller a route request upon consumer interest FIB match failure; and $H.CR$ corresponds to consumer to producer interests. Inequation 4.4 shows the lower bound expression for CRoS-NDN data delivery efficiency.

$$DDE_{CRoS-NDN} >= \frac{CR}{(L(2.KR + TR) + H(N.TR + AR + CR(FF + 1)))} \tag{4.4}$$

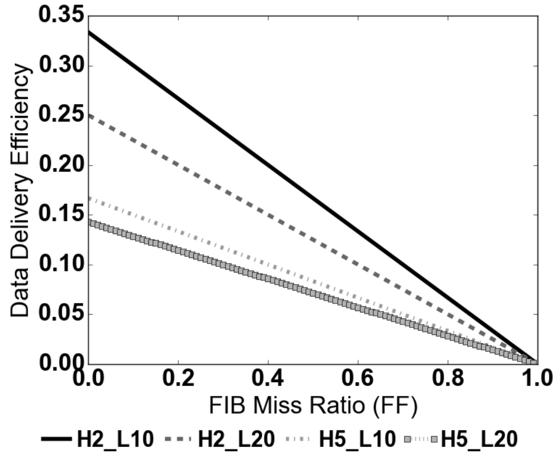
Figures 4.1, 4.2, and 4.3 present sample instances of the lower bound curves for the data delivery efficiency. These curves show, in a visual form, DDE comparative changes due to specific parameters.

Next, we derive upper bounds expressions for data delivery delay DDD for all the analyzed schemes. Data delivery delay is another important performance parameter that corresponds to the delay between consumer content request and consumer content reception. The DDD parameter, see Table 4.3, sums three delay components: CD - delay between consumer interest dispatch and content reception; AD - delay between producer content prefix announcement and network wide reach; and TD - delay between a topology change and network forwarding rules convergence. In worst case, the routing scheme converges upon any topology change adding TD, afterwards producer can announce its content AD, and finally consumer can ask the content CD. However, not all routing schemes pass through these three phases and, then, DDD components equals zero in some cases. The maximum **R**ound **T**rip **D**elay, RTD, between any pair of routers equals the diameter delay $RTD = 2.H.LD$. It is worth to that, in scenario without cache, the optimum DDD equals RTD.

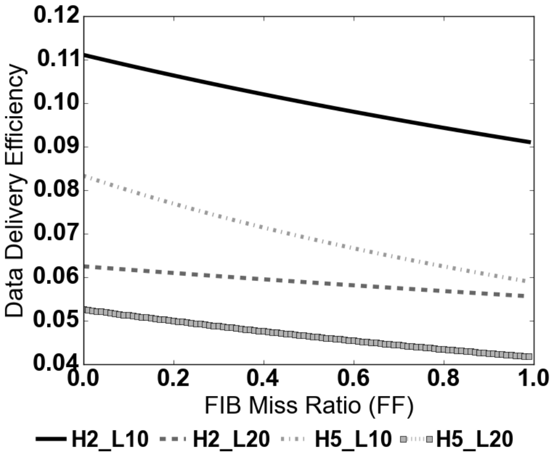
The CD component considers the round trip delay between consumer and producer for all schemes, except CRoS-NDN. In worst case, CRoS-NDN consumer first asks the controller a new route to content producer and, then, this additional procedure adds the round trip delay between consumer and controller. Inequation 4.5



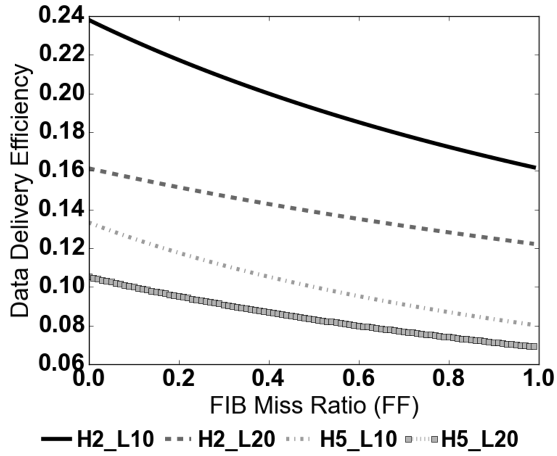
(a) ARPLike.



(b) OSPFLike.



(c) NLSRLike.

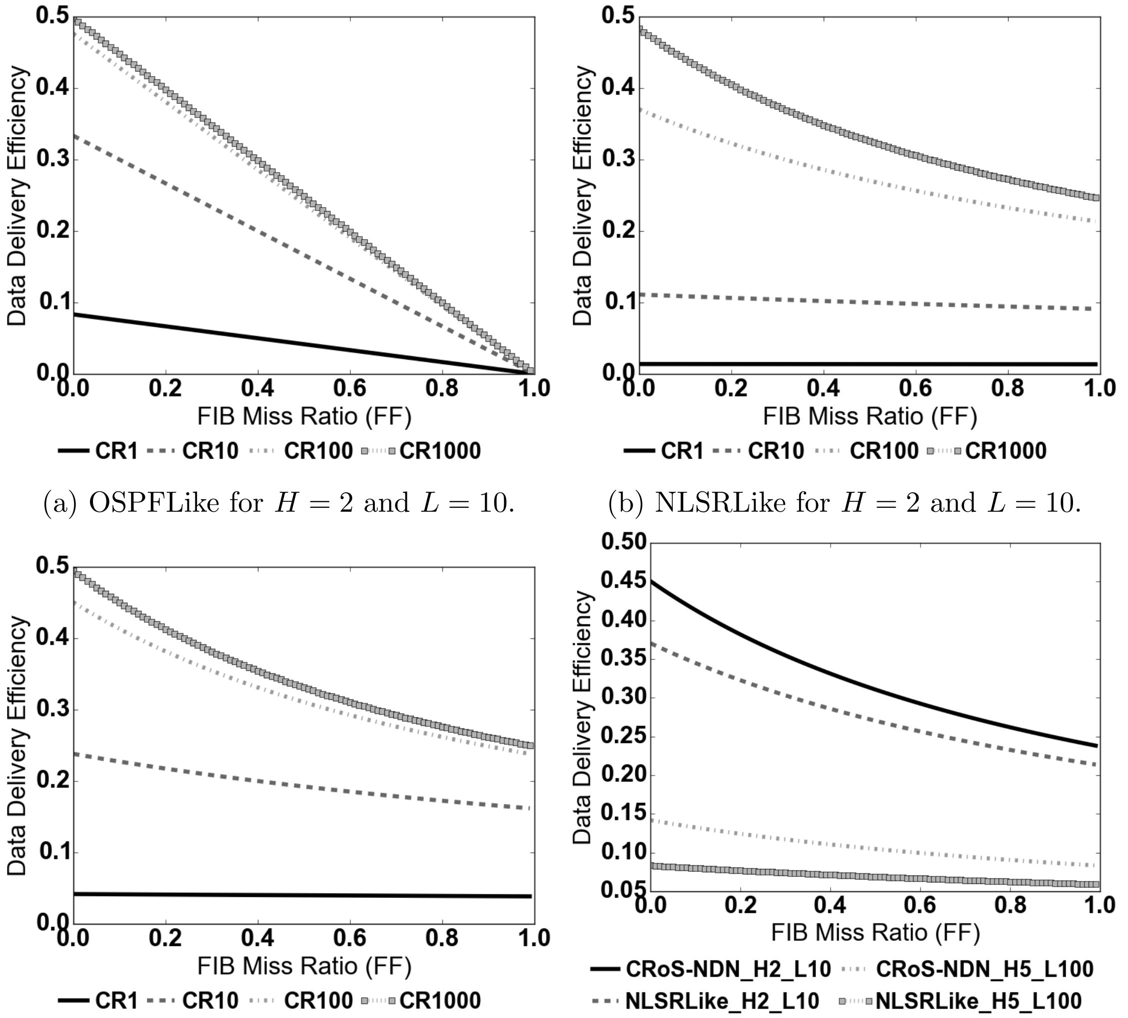


(d) CRoS-NDN.

Figure 4.1: Lower bound curves for Data Delivery Efficiency(DDE) with parameters: H in $\{2, 5\}$, L in $\{10, 20\}$, $CR = 10$, $KR = 1$, $AP = 1$, $AR = 1$, $TR = 0$, and $N = 10$.

Table 4.3: Upper bound expressions for Data Delivery Delay (DDD) components: $DDD = CD + AD + TD$. Consumer-producer Delay (CD), Announcement Delay (AD), Topology-update Delay (TD).

Scheme	CD	AD	TD
ARPLike	RTD	0	0
OSPFLike	RTD	$^{RTD}/_{2}$	$^{1}/_{KR}$
NLSRLike	RTD	$5.RTD + {}^{H}/_{KR}$	$4.RTD + {}^{H}/_{KR}$
CRoS-NDN	$2.RTD$	$^{RTD}/_{2}$	$^{3.RTD}/_{2} + {}^{1}/_{KR}$



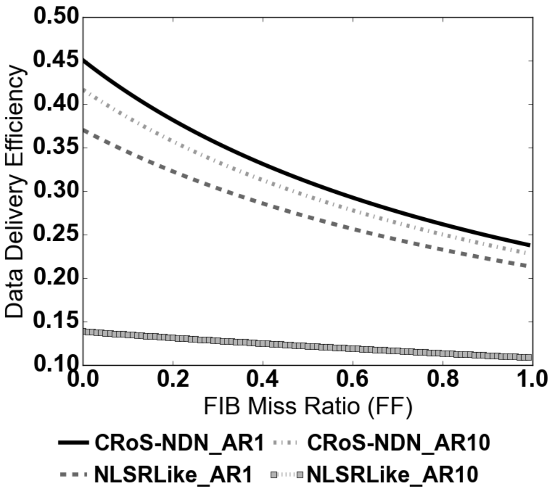
(a) OSPFLike for $H = 2$ and $L = 10$.

(b) NLSRLike for $H = 2$ and $L = 10$.

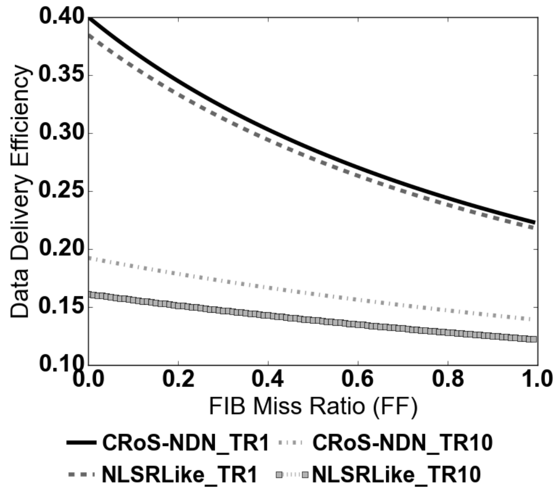
(c) CRoS-NDN for $H = 2$ and $L = 10$.

(d) $CR = 100$.

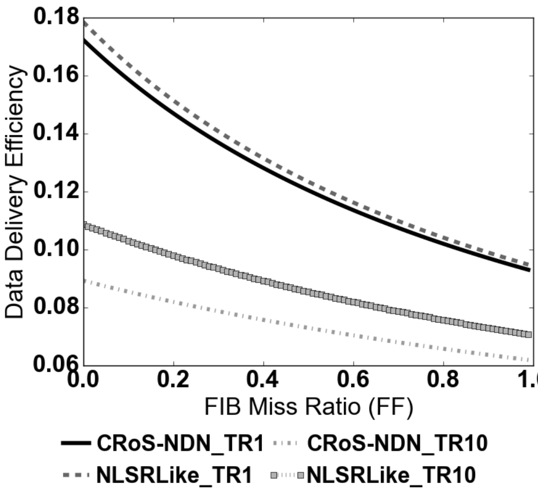
Figure 4.2: Lower bound curves for Data Delivery Efficiency(DDE) with parameters: H in $\{2, 5\}$, L in $\{10, 100\}$, CR in $\{1, 10, 100, 1000\}$, $KR = 1$, $AP = 1$, $AR = 1$, $TR = 0$, and $N = 10$.



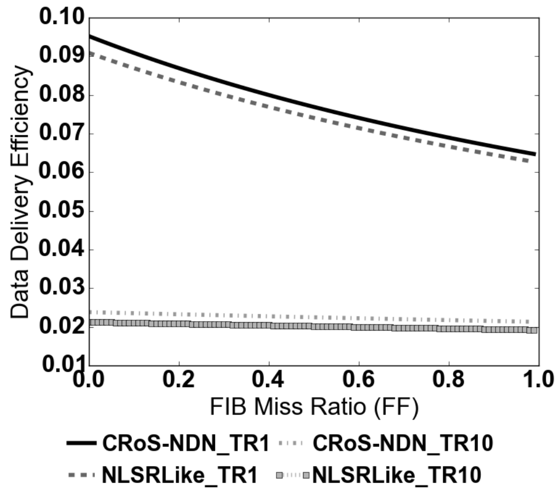
(a) AR in $\{1, 10\}$ for $TR = 0$, $H = 2$, $L = 10$, and $N = 10$.



(b) TR in $\{1, 10\}$ for $AR = 0$, $H = 2$, $L = 10$, and $N = 10$.



(c) TR in $\{1, 10\}$ for $AR = 0$, $H = 5$, $L = 10$, and $N = 10$.



(d) TR in $\{1, 10\}$ for $AR = 0$, $H = 5$, $L = 100$, and $N = 50$.

Figure 4.3: CRoS-NDN and NLSRLike lower-bound curves for Data Delivery Efficiency(DDE) with parameters: H in $\{2, 5\}$, L in$\{10, 100\}$, $CR = 100$, $KR = 1$, AR in $\{1, 10\}$, TR in $\{1, 10\}$, and N in $\{10, 50\}$.

shows the upper bound expressions for the CD component of data delivery delay.

$$CD_1 <= 2.H.LD = RTD \tag{4.5a}$$
$$1_{ARPLike,OSPFLike,NLSRLike}$$
$$CD_{CRoS-NDN} <= 4.H.LD = 2.RTD \tag{4.5b}$$

The AD component affects only the schemes that producer proactively announces content prefixes. ARPLike does not announce prefix and AD equals zero. OSPFLike and CRoS-NDN prefix announcement, respectively, adds to AD the one way producer to consumer delay and the one way producer to controller delay. NLSRLike prefix announcement employs the database synchronizing scheme. For each hop in the path from producer to consumer, NLSRLike adds to AD the LSDB hash exchange interval $1/KR$ and the prefix-LSA exchange delay. The prefix-LSA exchange employs five request and response sequential iterations and, then, it sums the delay $10.LD = 5.RTD$. The exchange of branch hashes takes four iterations with the four components of LSA name and the LSA exchange takes one additional iteration. Inequation 4.6 shows the upper bound expressions for the AD component of data delivery delay.

$$AD_{ARPLike} <= 0 \tag{4.6a}$$
$$AD_2 <= H.LD = \frac{RTD}{2} \tag{4.6b}$$
$$2_{OSPFLike,CRoS-NDN}$$
$$AD_{NLSRLike} <= H(10.LD + \frac{1}{KR}) = 5.RTD + \frac{H}{KR} \tag{4.6c}$$

The TD component affects only schemes that routers monitor network topology changes. ARPLike does not monitor topology changes and TD equals zero. Although OSPFLike routers do not monitor topology, prefix announcement periodic interval delays new paths convergence and it adds $1/KR$ to TD. NLSRLike routers update their local LSDB with a new neighbor-LSA upon local topology change. The LSDB synchronism for neighbor-LSA is one iteration faster than for prefix-LSA, because neighbor-LSA name has three components. CRoS-NDN router periodically monitors connectivity to its neighbors at interval $1/KR$ adding this value to TD. Additionally, topology changes can incur changes in path from router to controller. In this case, CRoS-NDN router needs to search a new path to controller and to re-register in controller. Controller discovery adds the router to controller round trip delay and the register renewal adds another router to controller one way delay to TD. Inequation 4.7 shows the upper bound expressions for the TD component of data delivery delay.

$$TD_{ARPLike} <= 0 \tag{4.7a}$$

$$TD_{OSPFLike} <= \frac{1}{KR} \tag{4.7b}$$

$$TD_{NLSRLike} <= H(8 \times LD + \frac{1}{KR}) = 4.RTD + \frac{H}{KR} \tag{4.7c}$$

$$TD_{CRoS-NDN} <= 3.H.LD + \frac{1}{KR} = \frac{3.RTD}{2} + \frac{1}{KR} \tag{4.7d}$$

Concerning the data delivery efficiency we can say that higher is the number of prefixes AP, better is CRoS-NDN and NLSRLike efficiency compared to OSPFLike. CRoS-NDN and NLSRLike only announce new prefixes with rate AR while OSPFLike periodically re-announces all prefixes AP with keep alive rate, KR. On the other hand, this OSPFLike comparative disadvantage reduces with the growth of topology change rate TR. CRoS-NDN shows a better efficiency than NLSRLike for scenarios with high number of prefix announcements. The efficiency decrease of our proposal CRoS-NDN is proportional to the prefix announcement rate and to the network diameter hops $H.AR$ while NLSRLike efficiency decrease is proportional to prefix announcement rate and to the network number of links $L.AR$. Furthermore, higher is the rate of interests for prefixes not installed in FIB $CR.FF$, better is CRoS-NDN efficiency compared to ARPLike. ARPLike floods interests without FIB match and the efficiency decreases proportionally to the number of links $L.CR.FF$. Unlike ARPLike, CRoS-NDN efficiency decreases proportionally to network diameter hops $H.AR + H.CR.FF$, $H.AR$ interest hops to producer register the content in controller and $H.CR.FF$ interest hops for consumer to request new routes from controller.

Concerning the Data Delivery Delay DDD depends directly on three parameters: network diameter in **H**ops, H, **L**ink **D**elay LD, and the keep-alive rate, KR. Lower is KR, higher is DDD for OSPFLike, NLSRLike and CRoS-NDN strategies. In special, for ($^1/_{KR} >> LD$), LD element becomes negligible. Then, ARPLike delay tends to 0, OSPFLike delay tends to $^1/_{KR}$, NLSRLike delay tends to $^H/_{KR}$, and CRoS-NDN delay tends to $^1/_{KR}$. Albeit smaller KR value reduces signaling overhead, it increases DDD delay for OSPFLike, CRoS-NDN, and NLSRLike.

Chapter 5

Simulation Results

In the preceding chapter, we analytically derived lower bounds for data delivery efficiency of NDN routing schemes and upper bounds for the respective data delivery delay. In this chapter, we present the simulator, the simulation results of the proposed Crontroller-based Routing Scheme for Named-Data Networking (CRoS-NDN), and comparative results of our CRoS-NDN proposal with the presented Distributed Routing Schemes for Named-Data Networking. We obtain performance simulation results for different scenarios that offers more detailed information of the analyzed schemes behavior.

5.1 ndnSIM Simulator

We have implemented our proposed scheme and the compared ones in the ndnSIM [57] simulator. To the best of our knowledge, ndnSIM is the closest to reality tool for NDN simulation. The ndnSIM reproduces the NDN model with a customizable forwarding strategy. Interest and data packets flow from node to node, and from/to node to/from application through faces. The strategy layer exposes customizable decisions on packet forwarding events. Thus, each routing scheme employs a specific forwarding strategy and specific applications to execute its procedures. Figure 5.1 shows the block diagram of the NDN node implementation on ndnSIM. We implemented two node modules to manipulate FIB and PIT entries based on data names: one executes specific forwarding strategy for each routing protocol and the other consumes/produces specific data packets related to specific routing scheme. The two modules employ internal calls to manipulate FIB, PIT, CS, and other state information.

In special, Figure 5.2 shows the block diagram of our CRoS-NDN router and controller implementation. CRoS-NDN controller and routers share the same structure, although their different functions in the proposed scheme. CRoS-NDN router implements a specific forwarding strategy and auxiliary applications to execute the scheme

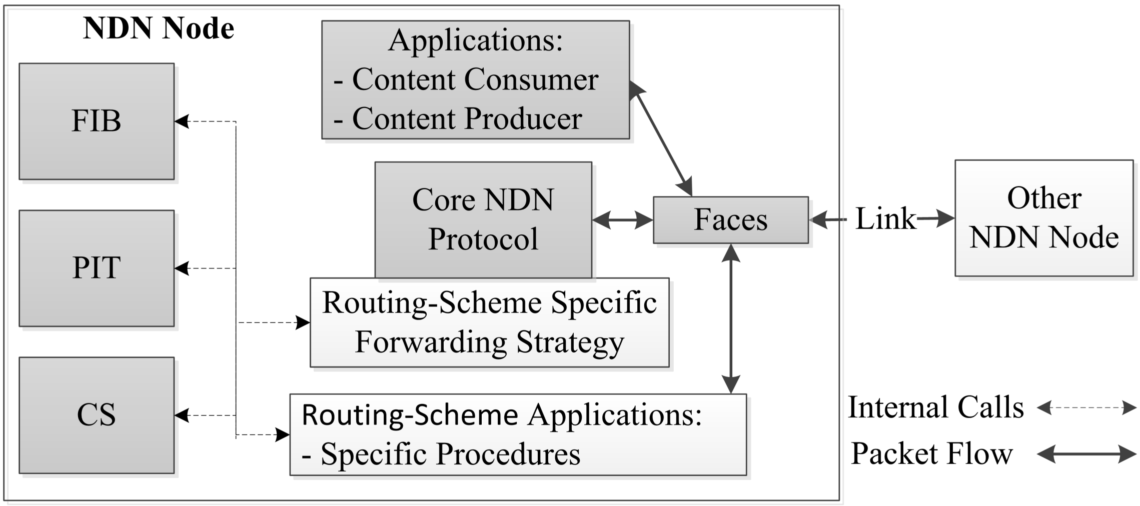


Figure 5.1: Customized ndnSIM node for implementing all routing schemes. The forwarding strategy module defines a specific routing scheme and interacts with specific applications module to manipulate FIB and PIT entries based on specific data names.

procedures. CRoS-NDN controller extends the NDN router with additional applications to implement the scheme procedures. CRoS-NDN router applications consume the controller produced data on the respective controller applications. CRoS-NDN forwarding strategy redirects interests for unknown prefixes to the Route Request procedure.

The simulation considers that multiple collocated nodes ran the CRoS-NDN controller function as a single entity and that these nodes share a database that stores both the named-data location and the routers adjacency information for a single domain. This assumption does not invalidate the performance evaluation because it relies on data-center well-connected infrastructure to host the nodes and, therefore, it eliminates processing power and storage bottlenecks of a single node.

5.2 CRoS-NDN Simulation Results

In this section we present CRoS-NDN simulations in order to demonstrate its operation and evaluate its performance. Figures 5.4 through 5.6, the first set of simulations, employ a small topology, see Figure 5.3, to easy the scenario construction and to explicit the result analysis. The second set of simulations, see Figures 5.7 through 5.13, extends the evaluation to other topologies. The simulation data employs 95% confidence interval. We present the maximum and the mean values of error bars in the legend of each plot when omitted in the curve.

The first set of simulations employs the specific topology with three different paths from consumer to producer, shown in Figure 5.3. The simulations evaluate

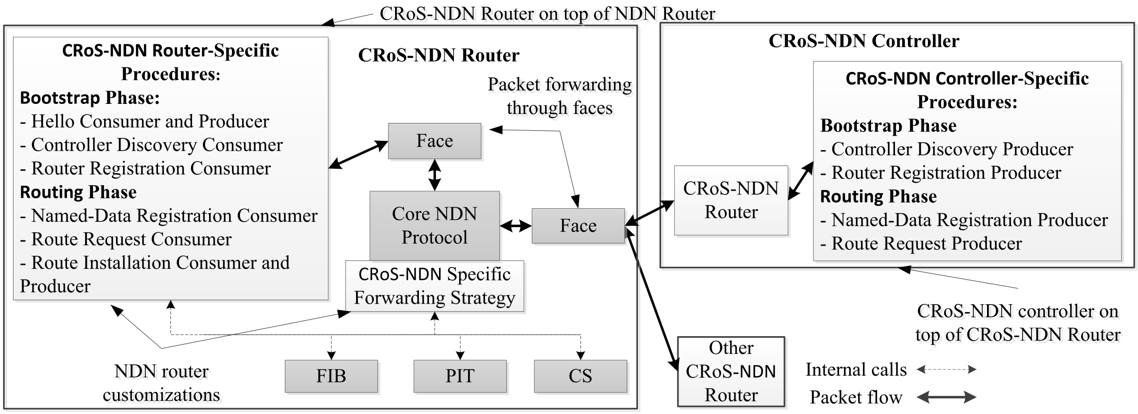


Figure 5.2: CRoS-NDN customization of NDN router. CRoS-NDN defines a specific forwarding strategy that interacts with specific procedures to manipulate FIB and PIT entries based on specific data names. CRoS-NDN controller employs additional procedures to execute the control plane on top of CRoS-NDN Router.

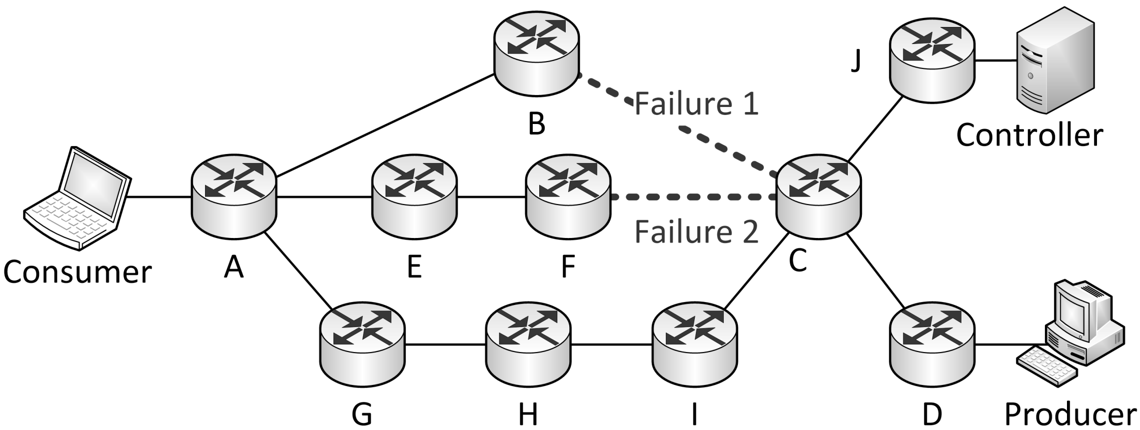
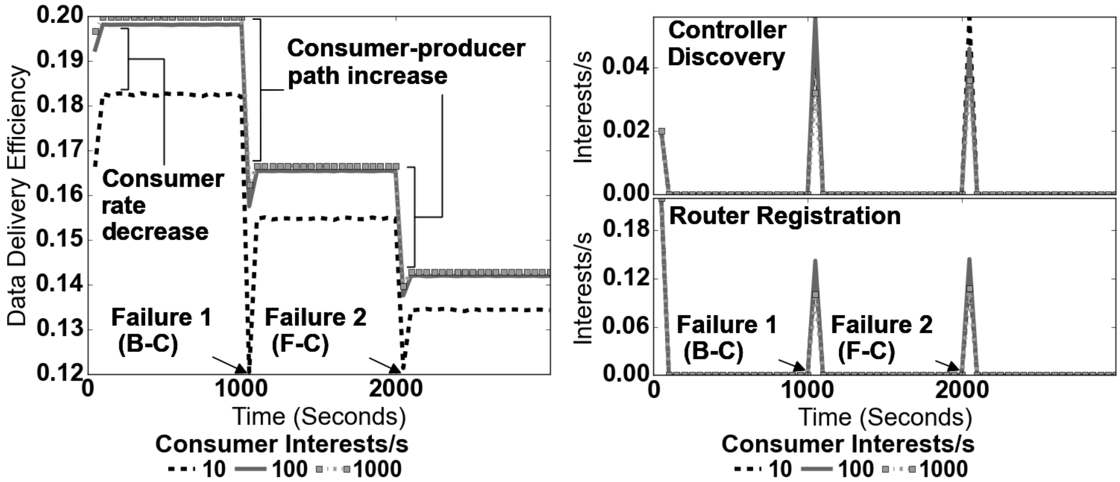


Figure 5.3: The topology used in the first set of simulations with three paths from consumer to producer: A-B-C-D, A-E-F-C-D, and A-G-H-I-C-D. The path distance increases after failures 1 and 2.

the proposed scheme operation after link failures and the consequent consumer-to-producer and router-to-controller path recovery. Beyond the self-discovery of controller and self-configuration of consumer-to-producer path, the distance between consumer and producer increases of one hop. The results confirm that CRoS-NDN self-discovery and self-configuration properties hold at start up and after topology changes.

CRoS-NDN assures that routers autonomously find the controller and configure/update the controller control plane. This feature avoids the manual configuration of nodes. Our proposal efficiently attains autonomy by restricting network flooding to initially find the controller, and on router to controller request failures. Figure 5.4a presents the data delivery efficiency and Figure 5.4b presents the controller-received interest rate for Controller Discovery (top graph) and for Router



(a) Data delivery efficiency. Max and mean error: 0.0064 and 0.0002.

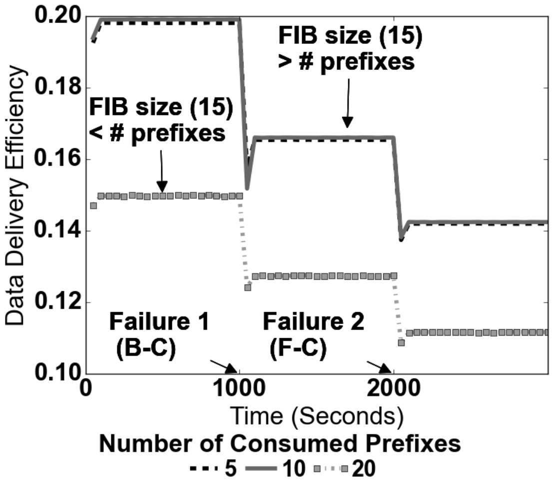
(b) Controller received interests. Max and mean error: 0.0154 and 0.0004.

Figure 5.4: a) The data delivery efficiency, b - top graph) the rate of interests received by the controller for the Controller Discovery procedure, and b - bottom graph) the rate of interests received by the controller for the Router Registration procedure, for Figure 5.3 topology and for consumer rates of 10, 100, and 1000 interests per second.

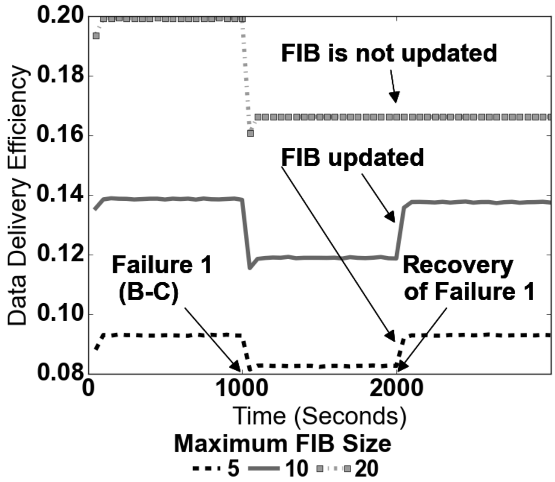
Registration (bottom graph) procedures for the consumer interest rates: 10, 100, and 1000 interests per second, using the topology in Figure 5.3. B-C and F-C links fail at 1000 and 2000 seconds respectively, `Failure 1` and `Failure 2`, and each failure adds one hop to the consumer to producer distance.

The simulation demonstrates the higher is the rate of consumer interests (10, 100, and 1000 per second), the closer is the data delivery efficiency to the optimum value $1/d$ (0.14, 0.19, and 0.20 before link failures), where d is the consumer to producer distance (5 hops before link failure). Furthermore, the simulation confirms the data delivery efficiency (0.20, 0.17, and 0.14 for 1000 consumer interests per second) is equal to $1/d$ (5, 6, and 7 hops) for the highest considered consumer interest rate and the efficiency decreases with lower consumer rates closer to Hello rate of 0.1 interests/s . Moreover, the simulation demonstrates the controller receives interests for Controller Discovery and Router Registration procedures only upon topology changes and, thus, these procedures show a low communication overhead. In addition, routers employ equal interests for the Controller Discovery procedure and, then, these interests are aggregated. The Router Registration interests are not aggregated because each router employs different interest names and, then, the Router Registration procedure shows a ten times higher interest rate received by the controller at start up.

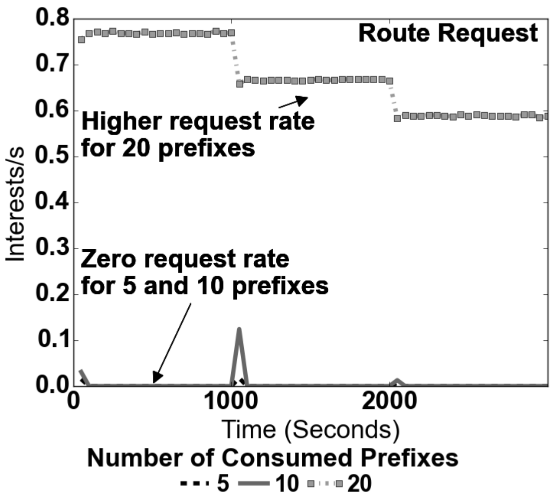
Figure 5.5a presents the reduction of the FIB memory requirement from the number of published prefixes to the number of simultaneous consumed prefixes.



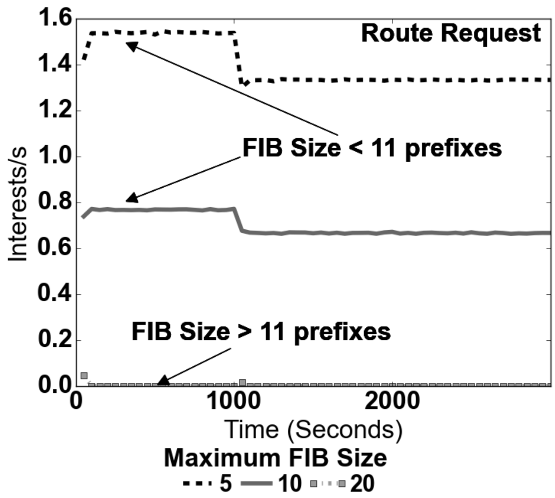
(a) FIB size 15. Max and mean error: 0.0060 and 0.0002.



(b) Consumed prefixes 11. Max and mean error: 0.0030 and 0.0002.



(c) Controller received interests in a). Max and mean error: 0.084 and 0.002.



(d) Controller received interests in b). Max and mean error: 0.051 and 0.004.

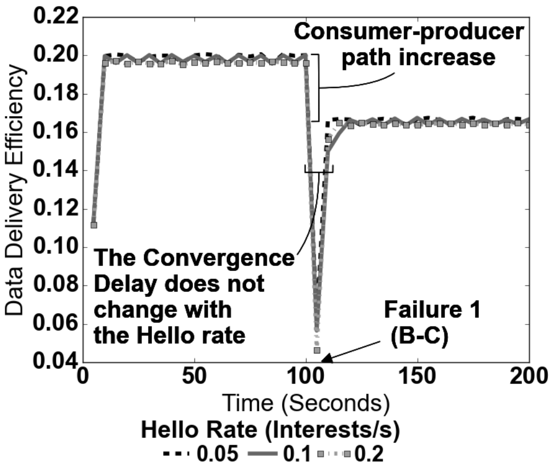
Figure 5.5: The number of prefixes and the FIB size simulations for Figure 5.3 network. a) The data delivery efficiency for 5, 10, and 20 simultaneous consumed prefixes and for FIB size of 15. b) The data delivery efficiency in a link failure/recovery event for 5, 10, and 20 FIB sizes and for 11 simultaneous consumed prefixes; Link B-C fails at 1000 seconds and recovers at 2000 seconds; The FIB entry replacement updates the data plane with the recovered link route at 2000 seconds. c) and d) The rate of interests received by the controller for the Route Request procedure of simulations a) and b), respectively.

CRoS-NDN router reduces FIB memory requirement because it adds new routes replacing the oldest added FIB entries. The simulation demonstrates that the data delivery efficiency does not decrease with the number of published prefixes for FIB sizes higher than the number of simultaneous consumed prefixes. Moreover, the result shows that CRoS-NDN correctly operates under insufficient FIB memory for simultaneous consumed prefixes, but the data delivery efficiency decreases proportionally to the rate of route requests to controller, Figure 5.5c. It is worth to note that when the consumer-controller path increases, then the round trip delay for route requests also increases and, therefore, the rate of route requests decreases due to interest aggregation.

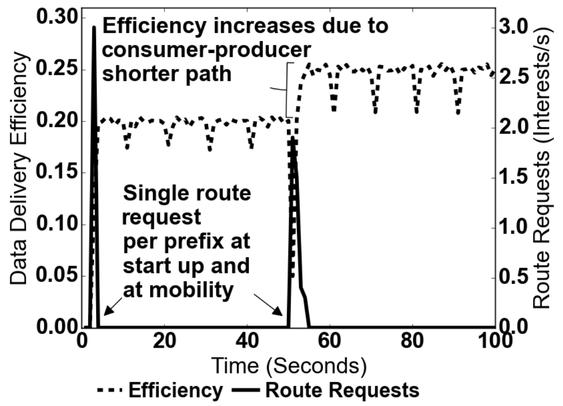
CRoS-NDN avoids frequent proactive FIB-updates of routers, and, then reduces the router-controller control-message overhead. It restricts the control messages to the number of unknown prefix of the requested data. Not all topology changes or content mobility require path updates. Actually, only the faults, identified by interest/data unbalance, that break the path from consumer to content requires path updates. Whenever a path breaks, the PIT entry expires, the router removes the respective FIB entry, and then the router requests a new route to the controller and updates its local forwarding information. Figure 5.5b presents the efficient separation of data and control planes. The simulation shows that when B-C link fails at 1000 seconds, the consumer to producer path increases one hop and the data delivery efficiency decreases proportionally. Afterwards, when B-C link recovers at 2000 seconds, the data delivery efficiency does not recover to the original value if the FIB size is higher than the number of simultaneous consumed prefixes. The simulation demonstrates that routers do not update their forwarding rules upon topology changes that do not break working paths. The approach avoids the proactive update of routers forwarding rules with the controller network view. Moreover, it is worth to note in Figures 5.5d that the route request rate does not change after link recovery because the consumer-controller path is not updated.

In the sequence, we evaluate the data delivery delay of the proposed routing scheme. The measures consider the convergence delay of the data delivery efficiency after topology changes. The Hello interest rate defines the detection latency of a link up/down change. Thus, the higher is the Hello interest rate, the lower is the latency of link change detection. On the other hand, the interest/data balance of Route Installation procedure accelerates the detection of connectivity failure between nodes in consumer to producer path and, then, this balance removes the delay dependency on the Hello procedure.

Figure 5.6a presents the convergence delay and the data delivery efficiency after a link failure for the Hello rate of 0.05, 0.10, and 0.20 Interests/s. The simulation shows the convergence delay does not change with the Hello rate due to route-



(a) Hello rate. Max and mean error: 0.038 and 0.002.



(b) Content mobility. Max and mean error: 0.015 and 0.006 (DDE), 0.77 and 0.02 (Route Requests).

Figure 5.6: The Hello interest rate and the content mobility simulations for Figure 5.3 network. a) The convergence delay in link failure event does not change with the Hello interest rate (0.05, 0.10, and 0.20 interests/s) due to route-install detection of connectivity failure. b) Route installation with a single route-request to controller per prefix and the data delivery efficiency for a mobile producer with 3 prefixes; The producer starts connected to router D and moves to router F at 50 seconds.

install detection of connectivity failure. The data delivery efficiency does not show significant change with Hello rate in this setting. Moreover, B-C link fails at 100 seconds, the failure adds one hop on consumer to producer distance, and reduces 0.03 on the data delivery efficiency, from a maximum of 0.20 before `Failure 1` to 0.17 afterwards. It is worth to note that `Failure 1` also changes the consumer to controller path. In this scenario, the route request after `Failure 1` fails without response and the Controller Discovery procedure starts. Therefore, the convergence delay sums the consumer-interest expiration delay, a first route-request expiration delay, the controller discovery delay, a second route-request delay, the delay of the route-install failure that detects the topology change, the router registration delay, a third route-request delay, and the consumer to producer delay through the new path. More importantly, these delay components depend on the link delay that is negligible compared to the interval between Hello interests.

The scheme avoids a request of route from each router in consumer-producer path to the controller. Figure 5.6b presents the Route Installation with a single route-request to controller per prefix, thus reducing the control message overhead. Producer node starts connected to router D and publishes 3 prefixes, then the producer moves to router F at 50 seconds. The simulation demonstrates that i) at start up, the controller receives 3 route requests, 1 per prefix, and ii) after producer mobility, producer registers the new data location and the controller receives

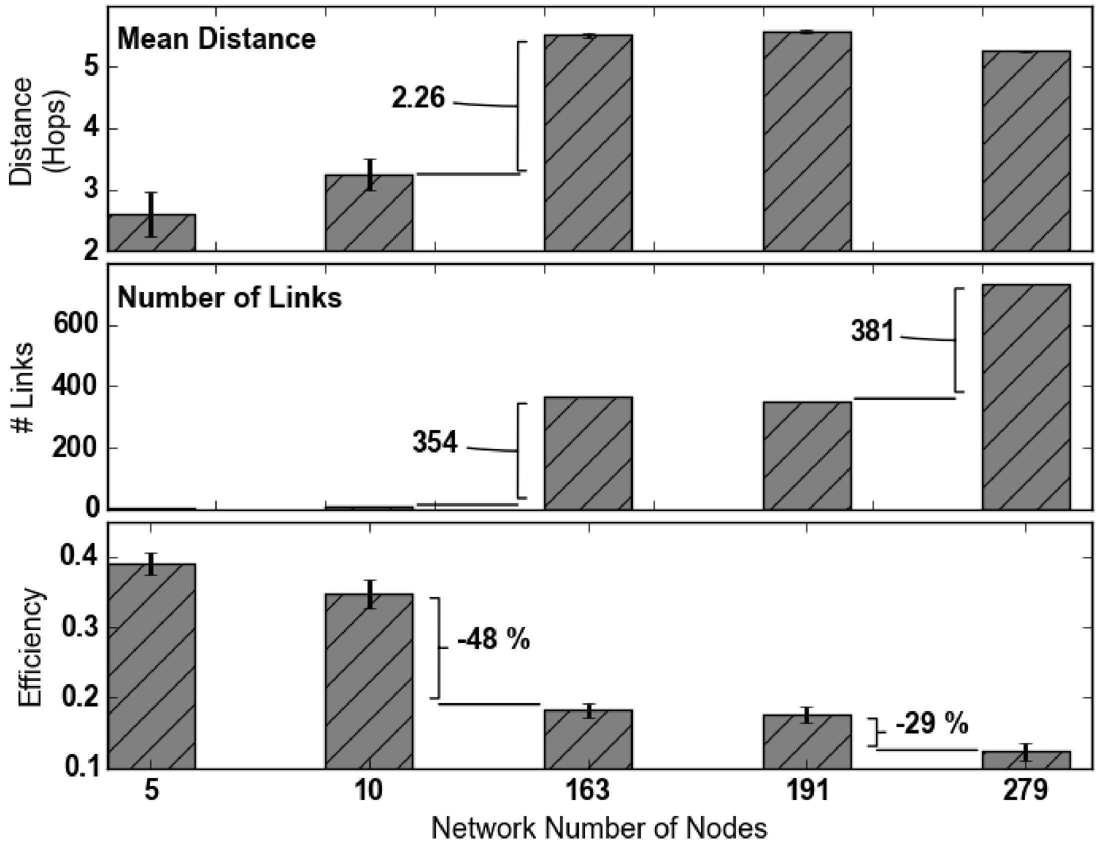
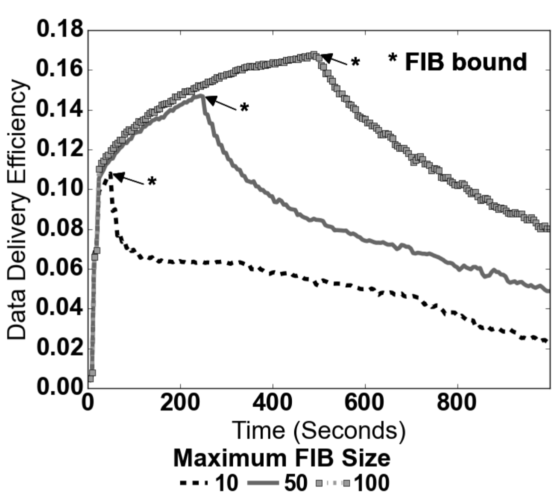


Figure 5.7: Comparative simulation for distinct network topologies. The network mean distance (top graph) for every combination of node pairs in each network: 2.60, 3.25, 5.51, 5.57, and 5.26 hops. The number of links (middle graph): 4, 12, 366, 350, and 731 links. The data delivery efficiency (bottom graph) for consumer, producer, and controller placed at random positions: 0.39, 0.35, 0.18, 0.18, and 0.12. The efficiency halves (−48%) when the consumer-producer distance doubles and the efficiency reduces less (−29%) when the number of links doubles.

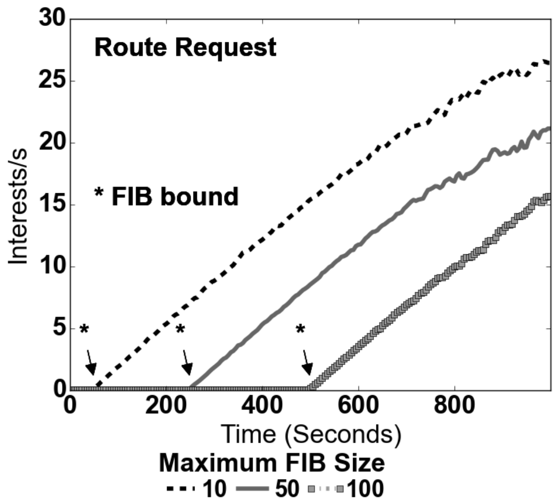
3 additional route requests. CRoS-NDN router reactively removes the failed routes pointing to the producer old location on PIT entry time-expiration and sends a new route request to controller upon a new consumer request for content. The number of route requests received by the controller does not change with the number of routers in consumer-producer path. Thus, for a single route request to controller, the controller replies with the data carrying the end-to-end path. Then, each router informs the new route to the next router on the path. It is worth to note that the convergence delay after content mobility, Figure 5.6b, is lower than the delay after topology change, Figure 5.6a. The convergence delay after content mobility sums the named-data registration delay, the expiration delay of the consumer-interest to content old location, the route request delay, and the delay of consumer to content new location.

In the sequence, we present the second set of simulations that extends the evaluation to other topologies. Figure 5.7 presents the data delivery efficiency in five different topologies, the topology in Figure 5.3 and four realistic network topologies. Like other works, the four ISP-like topologies are based on the largest connected component of Rocketfuel's VSNL, Ebone, Tiscali, and Telstra topologies, which corresponds to 5, 163, 191, and 279 nodes in Figure 5.7. Rocketfuel is a mapping technique that measures real router-level ISP topologies [9, 75]. We choose the topologies forming pairs with similar number of nodes, similar number of links, and similar mean distances to compare the effect of these parameters. We place at random positions the consumer, the producer, and the controller nodes in each simulation round. In the worst case, the consumer-producer distance is the network diameter. The mean distance, top graph, considers every combination of node pairs. The middle graph shows the number of links of the network. The data delivery efficiency, bottom graph, is close to the inverse of the mean distance. The results demonstrate that for a fixed consumer interest rate, the higher the number of links, the higher the number of Hello interests, and the lower the data delivery efficiency. Thus, the scheme presents a robust behavior, the efficiency does not decrease when the consumer interest rate grows with additional throughput of more links. Therefore, the scheme scales well for a controller with enough resources because the efficiency does not decrease due to additional messages. Furthermore, we envision that consumers can identify the distance to content, cache copies, and cooperatively register the copy location in the controller. Thus, the cooperation for specific contents potentially reduces the distance for new consumers and, this increases the data delivery efficiency for these contents with no cache in routers on the path to producer. This motivates an incrementally deployable approach for content producers irrespective of cache capacity in network routers.

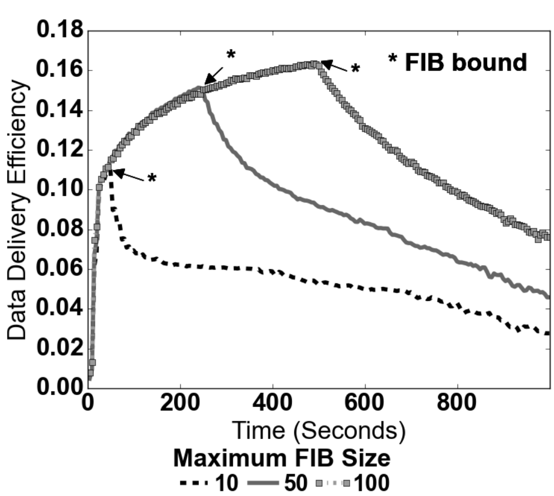
In the next simulation, Figure 5.8, we want to show the implications of unbalanced FIB memory capacity, link capacity, and amount of simultaneously consumed prefixes. We observe that the efficiency decreases due to recurrent route requests when the number of simultaneously consumed prefixes exceeds the FIB memory capacity. The scheme replaces FIB entries with a First-In First-Out (FIFO) policy. The early replacement of FIB entries causes the recurrent route requests in this scenario. Furthermore, the rate of route requests increases linearly with the rate of interests for prefixes without FIB entries up to link congestion. When increasing the rate of consumer interests beyond link congestion, the efficiency decreases and the growing rate of controller-received route requests also decreases due to interest retransmission caused by packet drop. It is worth to note that FIB entry removal erases all associated PIT entries and, therefore, the efficiency also decreases due to data packet drop when the FIB size is too small to store prefixes for time enough to receive producer data. The simulations employ Telstra and Ebone topologies with



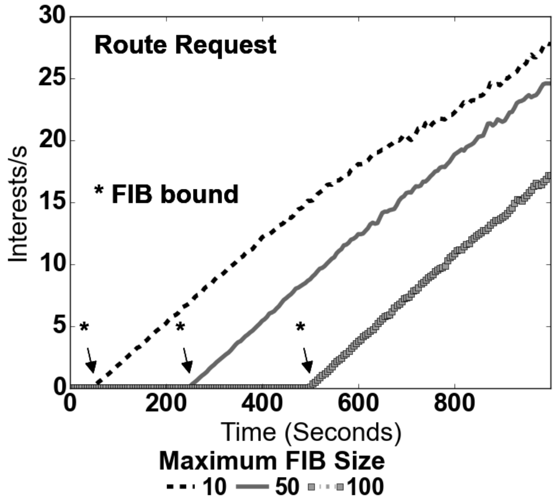
(a) Data delivery efficiency for Telstra topology. Max and mean error: 0.04 and 0.01.



(b) Controller received interests for Telstra topology. Max and mean error: 1.6 and 0.3.



(c) Data delivery efficiency for Ebone topology. Max and mean error: 0.03 and 0.01.



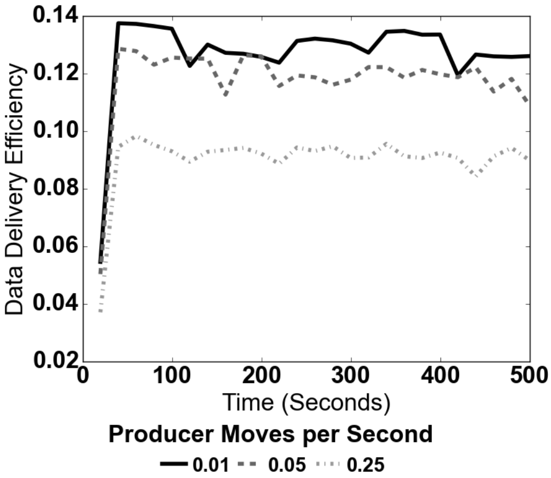
(d) Controller received interests for Ebone topology. Max and mean error: 1.4 and 0.3.

Figure 5.8: The data delivery efficiency, a) and c), and the rate of interests received by the controller in Route Request procedure, b) and d), for a growing interest rate and an increasing number of simultaneously consumed prefixes. A new consumer starts every 5 seconds and each consumer sends 1 interests/s for a distinct prefix. *FIB bound* arrows indicate when the number of simulatenously requested prefixes exceeds the FIB size capacity.

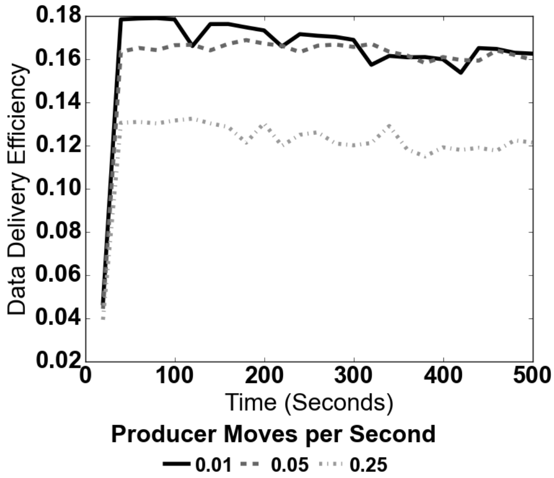
279 and 163 nodes respectively, a growing rate of simultaneously consumed prefixes, and an increasing rate of consumer interests. A new consumer starts every 5 seconds and each consumer sends 1 interests/s for a distinct prefix. We note that simulated consumers do not employ flow control to adapt the interest rate to response failures and, therefore, the growing rate of interests exacerbates the efficiency decrease.

In the sequence, Figure 5.9, we show the data delivery efficiency robustness with producer mobility and with the number of consumers. Whenever the producer moves, it starts the Register Named-Data procedure in order to inform its new localization. Moreover, consumer interests to old location expires and the Route Request procedure also starts. The higher is the rate of producer moves, the lower is the efficiency due to additional interests for these procedures. Furthermore, the higher is the number of consumers for a moving producer, the higher is amount of route requests due to producer moves. However, the efficiency improves with the number of consumers requesting equal content due to cache and interest aggregation. Therefore, our scheme shows a robust efficiency with content producer mobility and it avoids the consumption of router memory with prefixes in FIB for content that are not momently requested. The simulation employs 3 (30) consumers to request data with rate of 20 (2) interests per second in Figures 5.9a and 5.9c (5.9b and 5.9d), and a single producer to reply data packets. The producer moves with rates 0.01, 0.05, and 0.25 movements per second. We chose the simulation parameters in order to exhibit a reference efficiency behavior due to 10 times variation factor in both the ratio of consumer interests and producer moves, and the ratio of consumers and producer moves. Finally, we argue that our scheme can be integrated with the depot approach described by Zhang *et al.* [76]. The depot server intermediates consumer and producer communication and it keeps an updated route to producer avoiding that consumers have to request new routes when producer moves.

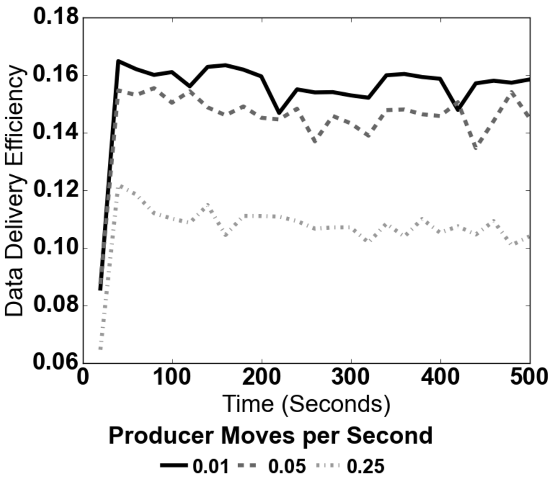
In the next simulations, we evaluate our proposal CRoS-NDN with consumers requesting content with a Zipf-Mandelbrot distribution for the prefix popularity. We consider constrained FIB memory, a growing rate of consumer interests, and short/long tail for the popularity distribution of content prefixes. We demonstrate that the efficiency decreases when the tail of the prefix popularity distribution increases and there is insufficient memory for the most of the available prefixes. In this case, the efficiency decreases for three reasons. The first reason is the higher rate of route request to controller that, in the worst case, halves the efficiency with one route request per consumer interest. The second reason relates to FIB and PIT association, an intrinsic characteristic of ndnSIM simulator that erases PIT entries on removal of corresponding FIB entry and, then, it leads to additional repeated interests from consumer for unanswered requests. The last reason is link congestion at higher consumer rates that, in the worst case, can congest the controller access



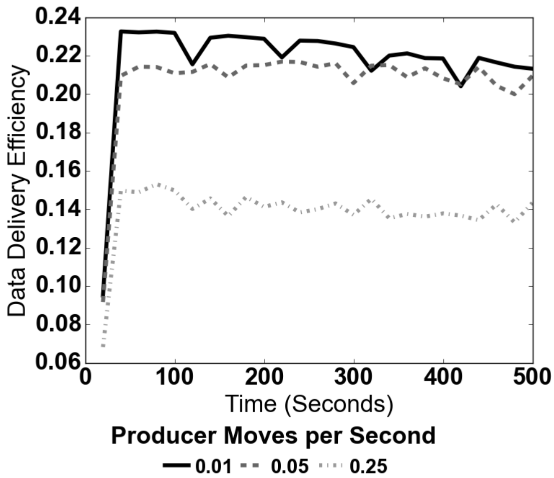
(a) Telstra topology, 3 consumers, and 20 interest/s per consumer. Max and mean error: 0.015 and 0.007.



(b) Telstra topology, 30 consumers, and 2 interest/s per consumer. Max and mean error: 0.012 and 0.005.

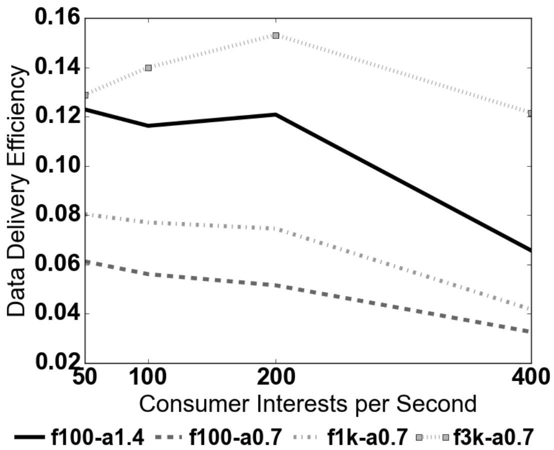


(c) Ebone topology, 3 consumers, and 20 interest/s per consumer. Max and mean error: 0.018 and 0.010

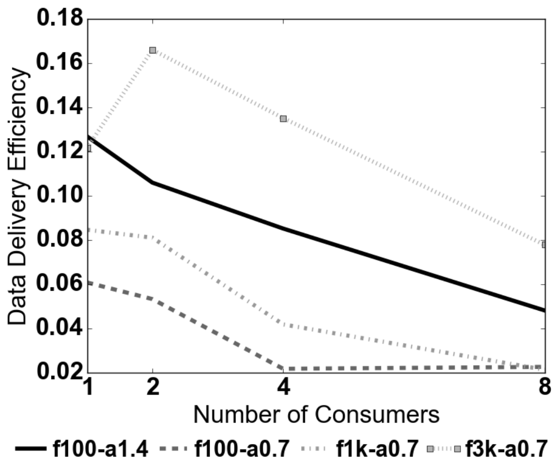


(d) Ebone topology, 30 consumers, and 2 interest/s per consumer. Max and mean error: 0.015 and 0.007

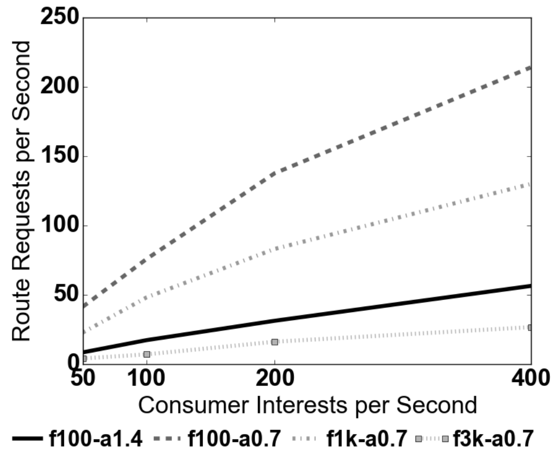
Figure 5.9: The data delivery efficiency for a growing rate of producer moves, distinct topologies, aggregated rate of 60 consumer interests/s, and 10 times variation in the number of consumers.



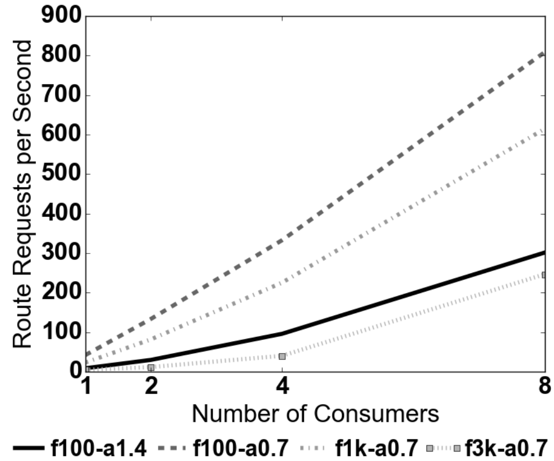
(a) Single consumer. Max and mean error: 0.029 and 0.021.



(b) Multiple consumers. Max and mean error: 0.025 and 0.018.



(c) Route requests received by the controller for single consumer. Max and mean error: 29 and 8.



(d) Route requests received by the controller for multiple consumers. Max and mean error: 69 and 16.

Figure 5.10: CRoS-NDN data delivery efficiency for consumer interests following the Zipf-Mandelbrot distribution for content prefix popularity. The simulation employs 3000 prefixes, FIB memory size of 100, 1000, 3000 entries (for f100, f1k, and f3k, respectively), and the Zipf α parameter values of 0.7 and 1.4 (for a0.7 and a1.4 respectively). Figures a and c consider a single consumer and a growing rate of consumer interests. Figures b and d consider multiple consumers and a fixed rate of 50 interests per second per consumer.

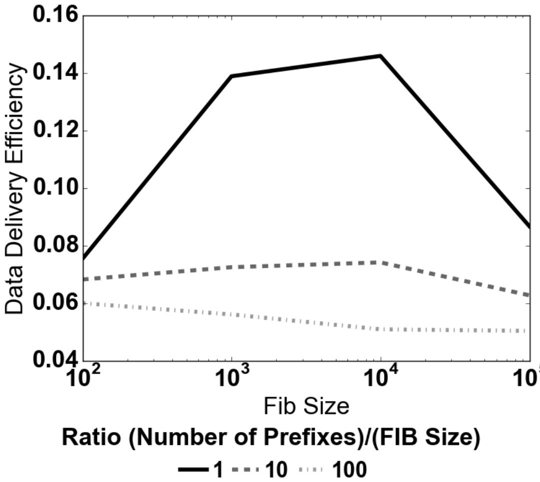
link and causes additional interest retransmission.

The longer is the tail of the prefix popularity distribution, *i.e.* lower α parameter of the Zipf distribution, the higher is both the rate of FIB match failures and the rate of route requests to controller when the FIB memory is insufficient for all content prefixes. Therefore, in consequence, the efficiency decreases. We choose the number of prefixes (3000), the FIB size (100, 1000, and 3000 entries), and the α values (0.7 and 1.4) [1] in order to explicit this behavior. Figure 5.10a shows the efficiency with a single consumer and a growing rate of consumer interests per second. Figure 5.10b shows the efficiency with a growing number of consumers and each consumer with a fixed rate of 50 interest per second. Figures 5.10c and d show the rate of route requests received by the controller for single consumer and multiple consumers cases, respectively. The higher rate of consumer interests causes higher rates of route requests. Furthermore, a higher number of consumers with the same prefix popularity distribution causes an aggregated prefix popularity distribution with longer tail, and, therefore, it decreases the efficiency due to a high rate of route requests. It is worth to observe that the aggregated rate of consumer interests is equal in Figures 5.10a, b, c and d. In addition, for small FIB size (100) and high rate of route requests, the FIB entry time in memory is lower than the round trip time and, thus, the early removal of a FIB entry and the associated PIT entries reduce the efficiency because of repeated route requests for the same prefix.

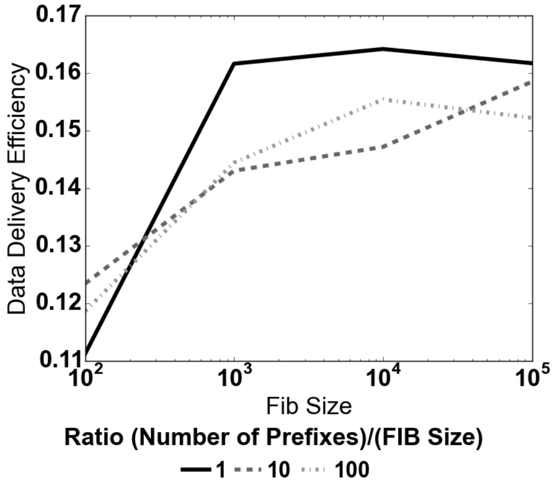
Figure 5.11 shows CRoS-NDN scalability and efficiency robustness for 3 orders of magnitude ratios of number of prefixes to FIB size. In addition, the results consider 4 orders of magnitude in the FIB size. In this scenario, a single consumer requests content with 100 interests per second. The higher the number of prefixes to FIB size ratio and the higher the Zipf α parameter, the lower the efficiency. It is worth to note that the higher is the number of prefixes, the lower is the ratio of requested prefixes to all prefixes considering a fixed time window and a fixed rate of consumer interests. Therefore, the efficiency decreases (stabilizes) for $\alpha = 0.7$ ($\alpha = 1.4$) with higher number of prefixes due to the limited simulation time.

Figures 5.10 and 5.11 point the CRoS-NDN potential bottleneck at the controller access link. The rate of route requests increases when there is insufficient FIB memory for the most of the solicited prefixes due to the long tail shape of prefix popularity distribution at core routers. In this scenario, the controller access link congests and causes interests retransmissions. The additional interests further reduce the efficiency. We plan to combine the Zhang *et al.* tunneling approach for NDN with our CRoS-NDN scheme to overcome this bottleneck [77]. We argue,

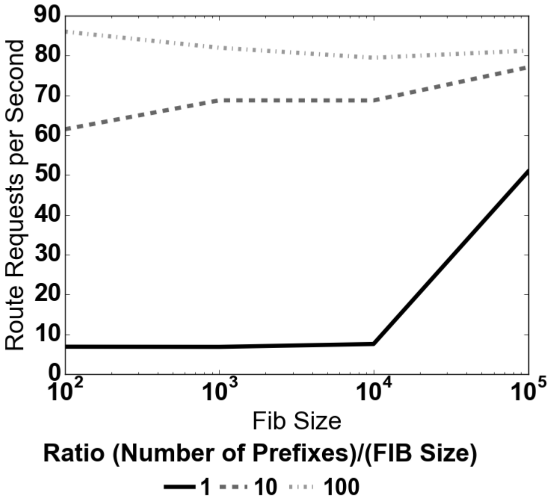
[1] We note that a single content is composed of multiples chunks that share a common name prefix. Therefore, the prefix popularity distribution has a lighter tail than the content popularity distribution [14].



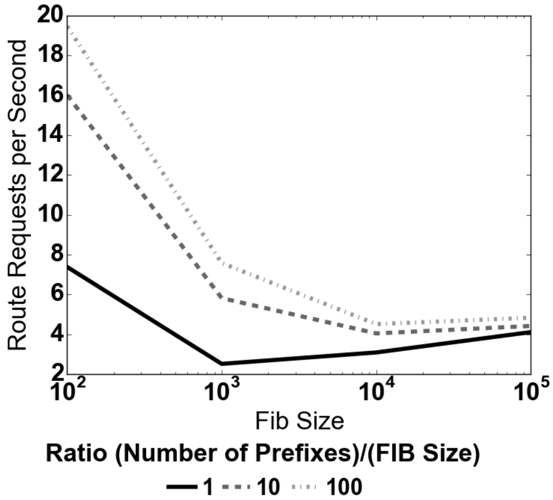
(a) Zipf $\alpha = 0.7$. Max and mean error: 0.011 and 0.009.



(b) Zipf $\alpha = 1.4$. Max and mean error: 0.018 and 0.016.



(c) Route requests received by the controller for Zipf $\alpha = 0.7$. Max and mean error: 7 and 4.



(d) Route requests received by the controller for Zipf $\alpha = 1.4$. Max and mean error: 2.6 and 0.5.

Figure 5.11: CRoS-NDN data delivery efficiency for the ratio of number of prefixes to FIB size. Consumer interests follow the Zipf-Mandelbrot distribution for content prefix popularity.

subject to further study, that the combined solution maintains our scheme features and reduces the rate of route requests to controller requiring less FIB memory at core routers for prefix popularity distributions with long tail at the network core, see Appendix A for more information.

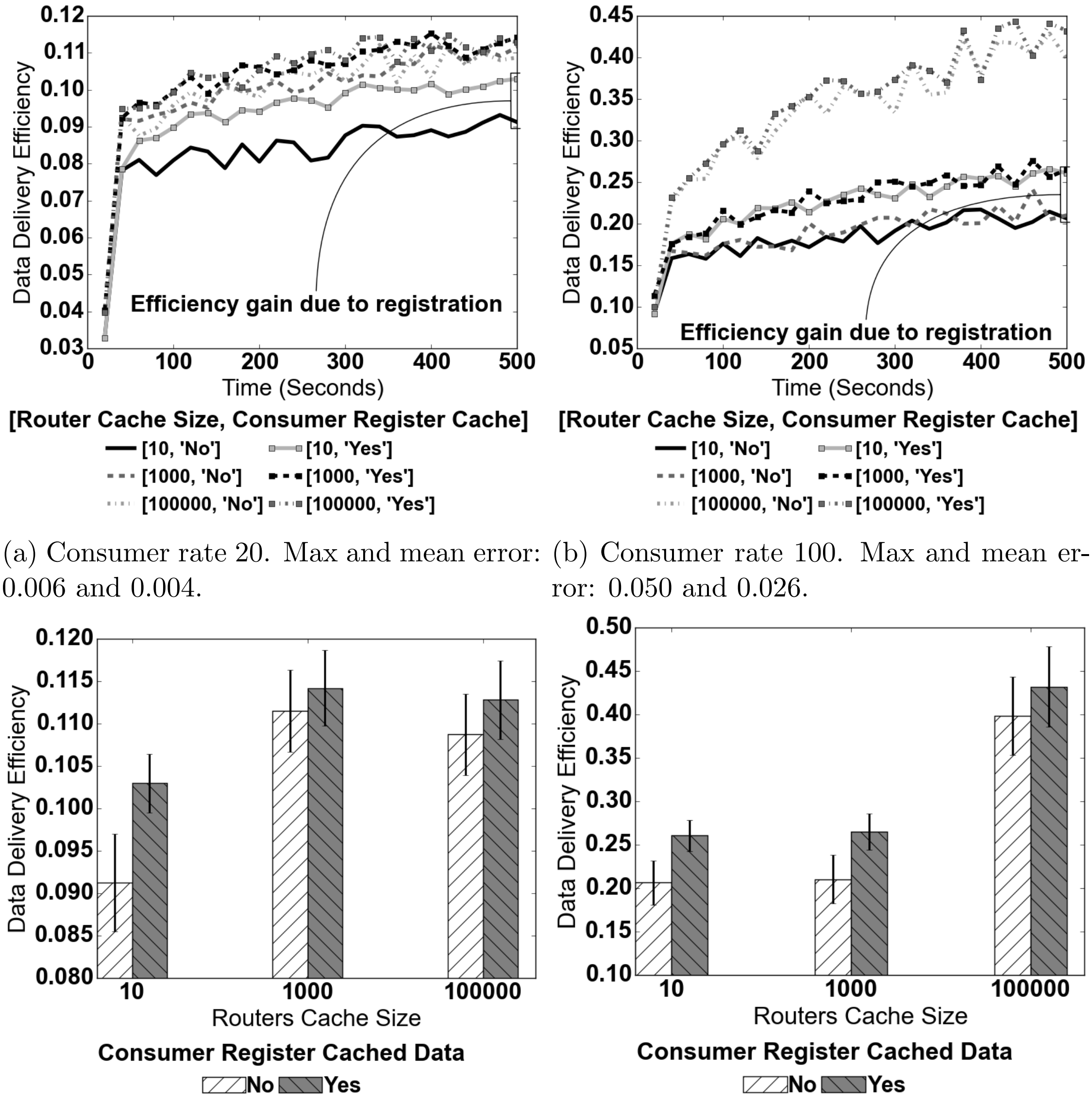
The preceding simulations showed the CRoS-NDN robustness with the amount of announced prefixes, simultaneously consumed prefixes, and the producer mobility rate. In the next set of simulations, we further demonstrate CRoS-NDN specific features and show that CRoS-NDN enables peer-to-peer and CDN functionalities over NDN with efficiency gains. We show that, with CRoS-NDN, it is worth to the global efficiency that consumers do register the cached copies of popular content [2]. Figure 5.12 compares the data delivery efficiency for the CRoS-NDN scheme with and without registration of content copies stored on consumers. Consumer nodes have unlimited cache capacity and routers have a limited cache capacity. Each consumer requests the same content sequence for 20 seconds and stops. A new consumer starts at every 20 seconds. In scenario with consumer registration of content copies, when the consumer stops, it registers the content copy location at the controller. The controller routes the interests to the closest registered copy [3].

The efficiency gain with consumer registration of content copies depends on router cache capacity and on the number of requested data. When routers have higher caching capacity than the requested data, registering content copies has no efficiency gain. Otherwise, when routers have smaller caching capacity than the requested data, registering content copies has a mensurable efficiency gain. Higher is the consumer interest rate, higher is the number of requested content items and higher is efficiency gain for the same cache size. Figures 5.12a and 5.12b compare the efficiency for consumer rates of 20 and 100 interests per second respectively. Figures 5.12c and 5.12d show the efficiency with error bars for the 25^o consumer in the same simulation. Additionally, the efficiency increases with the consumer rate because the Hello rate is fixed in 0.1 interests per second.

Figure 5.13 reinforces the efficiency gain of CRoS-NDN with registration of content copies stored on consumers over no registration of copies, when router cache size is insufficient for the requested data. Figure 5.13a shows that the highest efficiency gain occurs for the highest consumer rate (200) and a small cache size with 10 entries. Figure 5.13b shows no gain for consumer rate of 200 interests per second and a large cache size with 100,000 entries. Figures 5.13c and 5.13d show equivalent

[2]We note that NLSRLike and OSPFLike do not reach content copies outside the path to producer without additional prefix announcements. In addition, ARPLike do not announce content location. Therefore, we restrict this evaluation to CRoS-NDN.

[3]Like in BitTorrent, one can modify the strategy to distribute interests among the copies instead of sending interests just to the closest one. The BitTorrent strategy speed up the content transfer when consumer access link has higher capacity than the access link of each content copy individually.



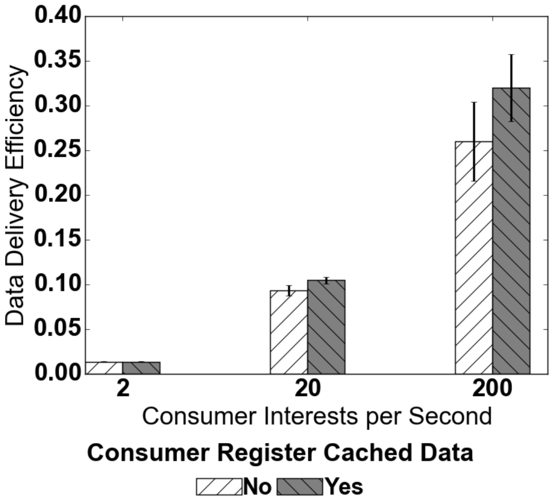
(a) Consumer rate 20. Max and mean error: 0.006 and 0.004.

(b) Consumer rate 100. Max and mean error: 0.050 and 0.026.

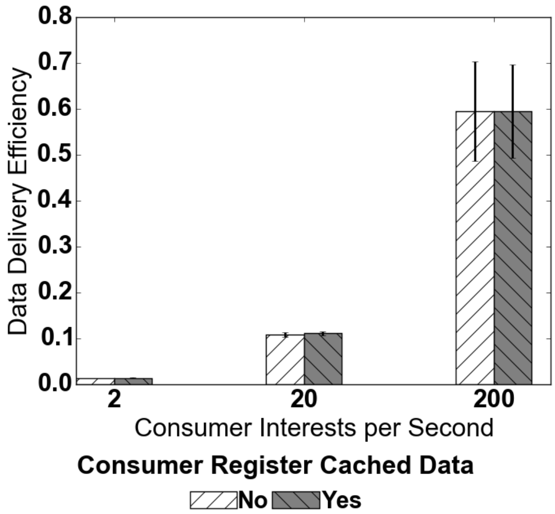
(c) Consumer rate 20.

(d) Consumer rate 100.

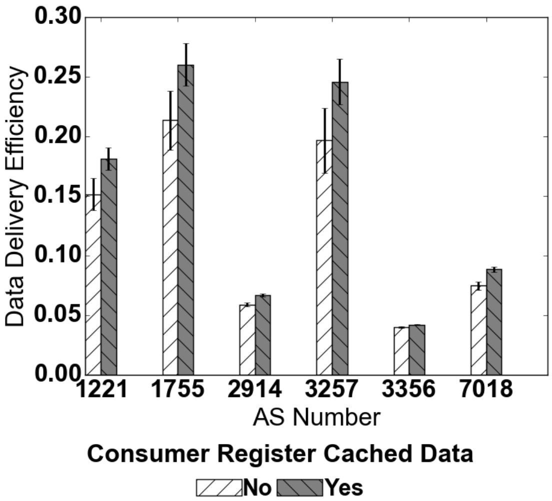
Figure 5.12: CRoS-NDN data delivery efficiency increase with consumer registration of data copies for consumer rates of 20 (figures a and c) and 100 (figures b and d) interests per second.



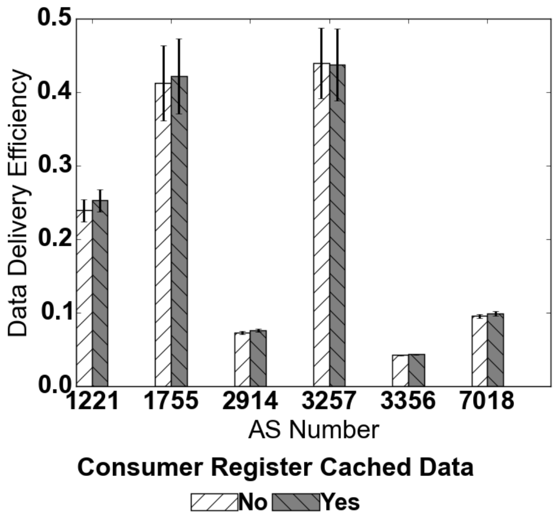
(a) Cache size 10.



(b) Cache size 100.000.



(c) Cache size 10.



(d) Cache size 100.000.

Figure 5.13: CRoS-NDN data delivery efficiency increase with consumer registration of data copies for cache sizes of 10 (figures a and c) and 100.000 (figures b and d).

results in different topologies for cache sizes of 10 and 100,000 entries respectively and for consumer rate of 100 interests per second.

Announcing content copies location allows consumers to reach a closer copy that is outside the path to the producer. CRoS-NDN shows a low overhead for the registration of content copies location. This is in opposition with OSPFLike and NLSRLike that shows poor performance when the rate of prefix announcements increase. In addition, real traffic presents a long tail distribution for the content popularity and the limited cache size of routers along the path to producer aggregates only repeated requests inside a small time window. We envision that the registration of content copies location is a potential solution for CDN over NDN. A router can proxy interest for specific prefixes and cache the respective data closer to potential consumers for longer time windows. Additionally, our scheme enables a form of peer-to-peer content distribution for NDN.

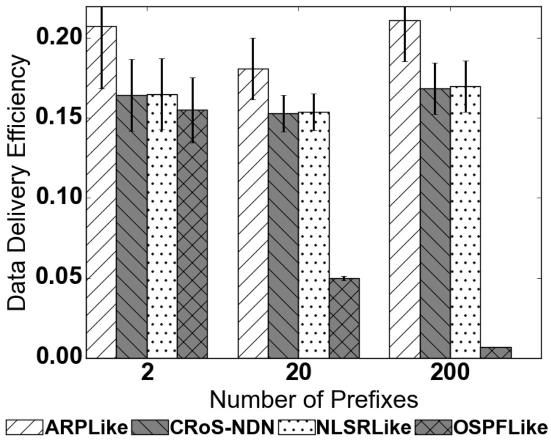
5.3 Comparative Simulation Results for CRoS-NDN and Distributed Routing Schemes

In this section, we present comparative simulations of our proposed CRoS-NDN scheme and the presented distributed routing schemes. When not stated in opposition, in each simulation round, consumer and producer routers are chosen randomly. The different distances from consumer to producer and from consumer to controller, in our CRoS-NDN proposal, cause the variation represented by the error bar in each plot. In addition, like other works, the simulations employ ISP-like topologies based on the largest connected component of Rocketfuel's topologies [9, 75], a mapping technique that measures real router-level ISP topologies. When not state in contrary, we employ the AS 1755 (Ebone) topology with 163 nodes and 366 links. It is worth to note that the network mean distance is 7.36 hops, the diameter is 22 hops, and the respective reference values for the data delivery efficiency are $DDE = 1/7.36 = 0.14$ for the mean case, and $DDE = 1/H = 0.05$ for the worst case. We choose the AS 1755 as the main topology because it has a sufficiently high number of links in comparison with diameter, $L >> H$, to reflect the flooding negative effect on efficiency. Furthermore, the keep-alive rate value KR is set to 0.1 for the OSPFLike periodic prefix announcement, like in OSPF [69], and for the NLSRLike/CRoS-NDN Hello procedure, like in NLSR [70]. We employ equal $KR = 0.1$ values in order to verify a fair comparison and we point that higher (lower) KR values decrease (increase) the efficiency and increase (decrease) the data delivery delay for these three schemes; however, different KR values do not change the comparative behavior with the increase in the number of prefixes. More importantly, we set conditioned values for simulation parameters in order to exhibit specific comparative results that would be obfuscated with real world values without any conditioning. Additionally, we emphasize the conditioning purpose is to explicit individual limitation factors of each scheme.

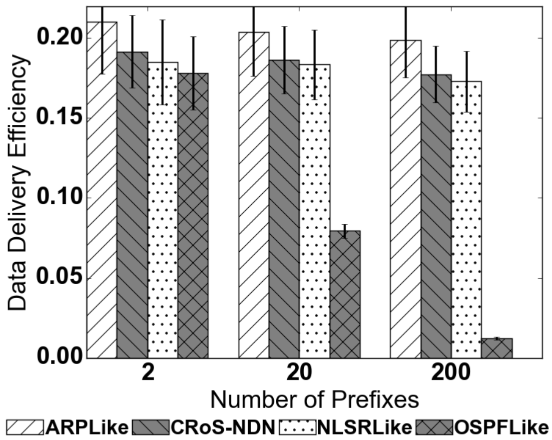
In the first set of simulation, we want to show the performance behavior of Data Delivery Efficiency when we increase the number[4] of prefixes by two orders of magnitude, from 2 to 200, and also when we restrict the FIB size. Figures 5.14a and b demonstrate the OSPFLike scalability weakness with the number of prefixes increase, even considering router with unlimited FIB memory. OSPFLike data delivery efficiency strongly decreases from 0.155 to 0.006 with the number of announced prefixes increase in Figure 5.14a, and from 0.178 to 0.012 in Figure 5.14b. The strong efficiency decrease of OSPFLike routing scheme is due to the periodically announcement of all available prefixes. It is worth to note that smaller KR values reduce the factor of OSPFLike efficiency decrease with the number of prefixes, but it does not change the tendency. On the other hand, ARPLike, NLSRLike, and our proposal CRoS-NDN efficiency shows very little variation with the number prefixes because these schemes avoid the periodic network flooding of available prefixes. The simulations of Figure 5.14a and b consider two consumers and each one requests sequential data for one distinct prefix with rate of 40 and 80 interests per second, respectively.

Figures 5.14c and d show the Data Delivery Efficiency behavior for constrained FIB memory. The results demonstrate the ARPLike scalability weakness with the increase of the number consumed prefixes beyond the FIB memory capacity. In Figures 5.14c, the simulation employs 15 simultaneously consumed prefixes, each one with 10 interests per second, 15 announced prefixes, and a growing number of FIB entries supported per router. We choose an amount of announced prefixes that smooths the OSPFLike weakness with prefix announcements and that shows the effect of FIB memory deficiency. Under FIB memory restriction, all routing schemes replace the oldest installed entries by the new ones (first-in first-out – FIFO). ARPLike efficiency suffers a lot by each FIB entry removal because it recurrently floods the network and, thus, the efficiency decreases proportionally to the number of network links. Unlike ARPLike, all the other schemes do not flood consumer interests upon FIB match failure. NLSRLike efficiency shows very little variation with the number supported FIB entries per router because NLSRLike router employs its local control plane to reinstall the forwarding rules on FIB. OSPFLike efficiency decreases, due to the lack of memory for part of prefixes, from 0.09 to 0.04 when the FIB memory reduces from 20 to 5 entries. CRoS-NDN efficiency also reduces from

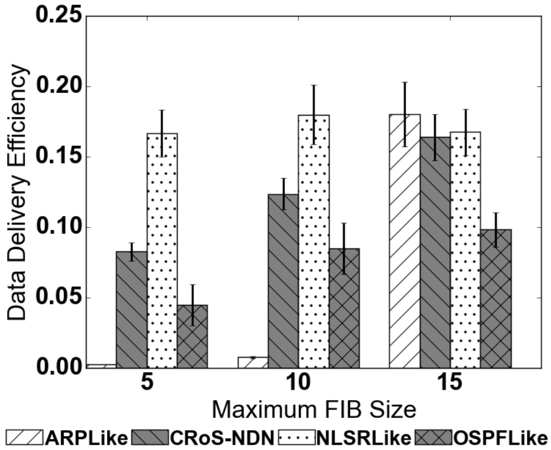
[4] We denote *consumed prefixes* the prefixes of content requested by consumers and we denote *announced prefixes* or simply *prefixes* the prefixes of content available at producers.



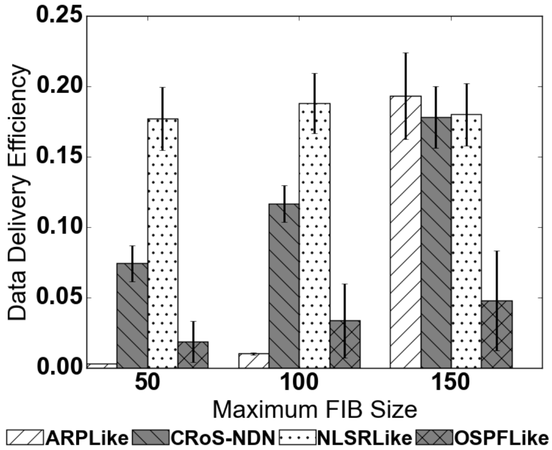
(a) OSPFLike efficiency decrease for 80 interests per second.



(b) OSPFLike efficiency decrease for 160 interests per second.



(c) ARPLike efficiency decrease for 15 consumed prefixes.

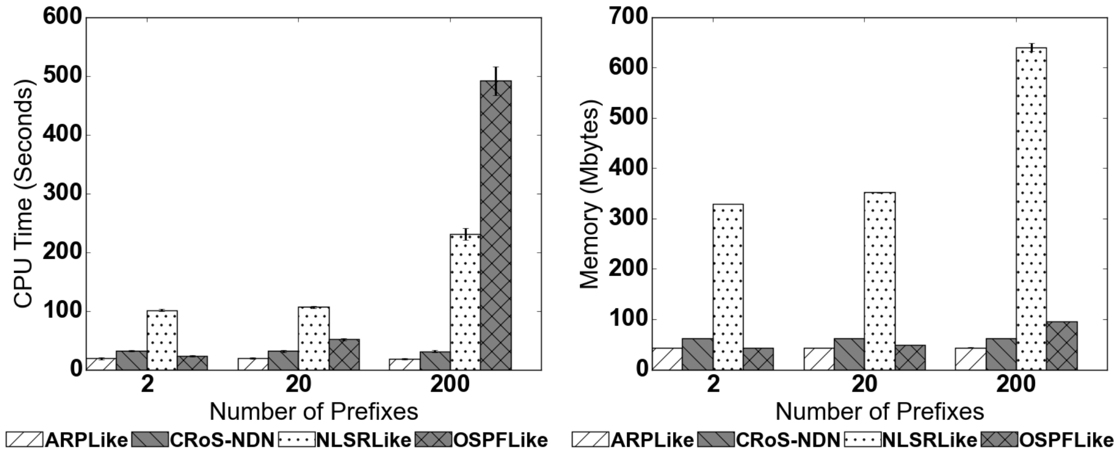


(d) ARPLike efficiency decrease for 150 consumed prefixes.

Figure 5.14: Data delivery efficiency for: a and b) unlimited FIB memory and a growing number of announced prefixes, and c and d) different FIB sizes and 15/150 simultaneous consumed prefixes.

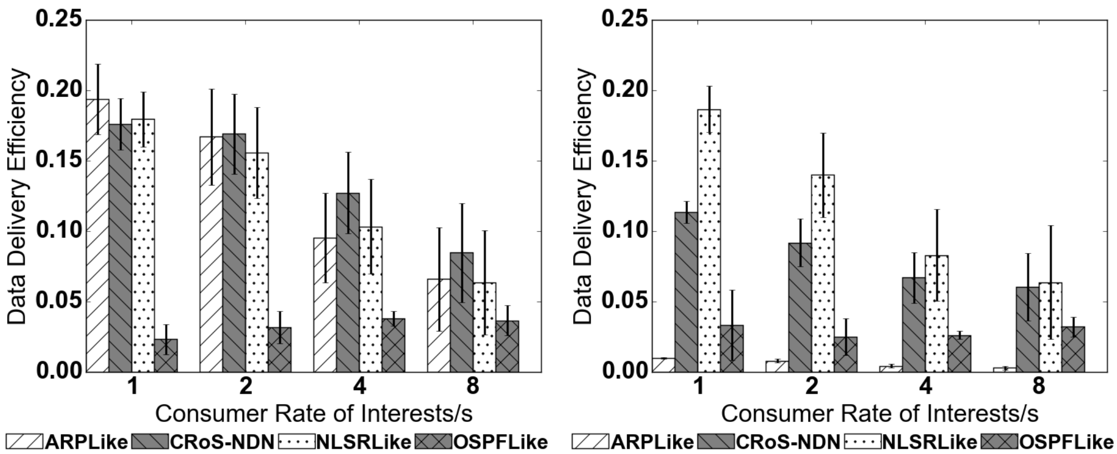
0.17 to 0.08, a reduction by a factor close to two that corresponds to the additional hop distance from consumer to controller for route requests. Consumer-producer and consumer-controller mean distances are equal to the network mean distance. In Figures 5.14d, the simulation employs 150 simultaneously consumed prefixes, each one with 1 interests per second, 150 announced prefixes, and a growing number of FIB entries supported per router. Figures 5.14c and d show similar results for a comparative ten times higher number of consumed prefixes and of FIB size, and a ten times lower rate of interests per prefix.

Figure 5.15 shows the processing time and the memory consumption of each simulation round for each scheme and for a growing number of prefixes. The results point the real consumed resources of our implementation and mirror the total consumption of resources by network routers of each scheme. NLSRLike and OSPFLike



(a) Processing time. (b) Used memory.

Figure 5.15: Processing time and memory consumption for each simulation round.



(a) Unlimited FIB memory. (b) FIB memory Size = 100.

Figure 5.16: Data delivery efficiency for a growing rate of consumer interests per prefix and 150 prefixes: a) unlimited FIB memory and b) limited FIB memory.

show the highest resource consumption. We note that, although the controller capacity does not scale infinitely, CRoS-NDN shows an economical use of resources for a growing number of prefixes.

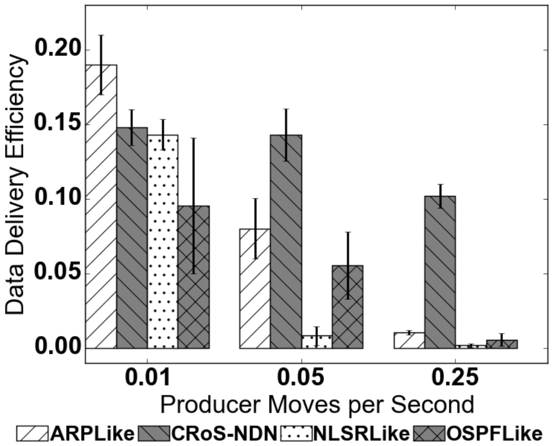
Figure 5.16 shows the the Data Delivery Efficiency for a growing rate of consumer interests per prefix. The efficiency decreases due to congestion of excessive requests above link capacity. The results reinforce OSPFLike low efficiency with the number of prefixes, 150. Additionally, Figure 5.16b shows ARPLike low efficiency with FIB memory smaller than the amount of simultaneous requested prefixes.

Multihoming and mobility is a great problem in today's Internet because of the semantics overload of IP. Our proposal is based on plane separation and, then, natively splits localization and identification. Therefore, in this second set of simulations, we show the Data Delivery Efficiency robustness to the content-producer mobility, an important feature of our proposal CRoS-NDN as depicted in Figure 5.17. Furthermore, we show the robustness of CRoS-NDN efficiency when increasing by one order of magnitude the number of announced prefixes and of consumers. In order to explicit the comparative efficiency trend, the simulation considers 3 consumers per announced prefix, each consumer sending 20 interests per second, unlimited FIB memory, and a growing rate of producer moves. Figures 5.17a and 5.17c present the data delivery efficiency for, respectively, 1 and 10 content prefixes in order to compare the combined effect of content mobility and the number of prefixes. In order to visualize the efficiency temporal evolution, Figures 5.17b and 5.17d show the curves for the rate of 0.05 producer moves per second of figures 5.17a and 5.17c, respectively. The results show that producer mobility increases both the ARPLike interest flooding for content search and the OSPFLike/NLSRLike announcements of producer prefixes and, thus, it strongly decreases the efficiency of these schemes. However, unlike for ARPLike, the growth of consumer interests rate with the number of prefixes contributes positively to OSPFLike and NLSRLike efficiency. On the other hand, CRoS-NDN presents the best results with fast convergence and low overhead for producer location update.

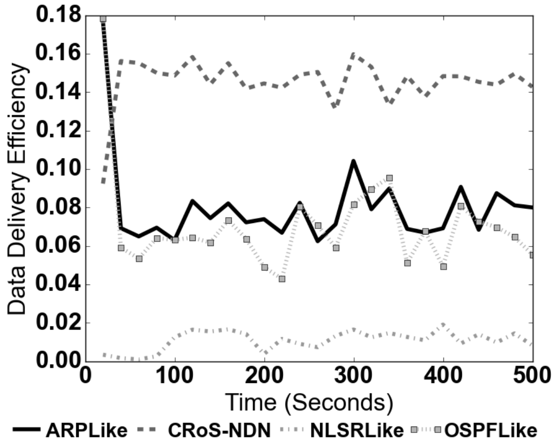
In the third set of simulations, we want to show the strong resiliency characteristic of our proposal that presents a fast start up and link-failure recovery. The data delivery efficiency DDE time evolution gives an indirect measure of convergence latency represented in the data delivery delay DDD metric. Figure 5.18a presents the latency for the convergence at start up and at the recovery from a link failure to a secondary longer path. CRoS-NDN presents a faster convergence delay because it only depends on routers delay to update its local information on the controller and routers delay to receive new routes from the controller. NLSRLike slower convergence is due to the hop by hop database synchronization latency. Furthermore, the set up convergence takes even longer due to the greater number of differences among routers databases. ARPLike and OSPFLike schemes show similar and small delay values because ARPLike immediately floods interests for unknown prefixes and OSPFLike convergence depends only on the producer prefix announcement arrival to install new routes.

In Figure 5.18b, we demonstrate the CRoS-NDN fast propagation of new routing information in comparison with NLSRLike. The producer announces one new prefix per second in the initial 100 seconds. The prefix announcement reduces the NLSRLike efficiency due to the database synchronization packets and, additionally, NLSRLike shows a higher convergence delay.

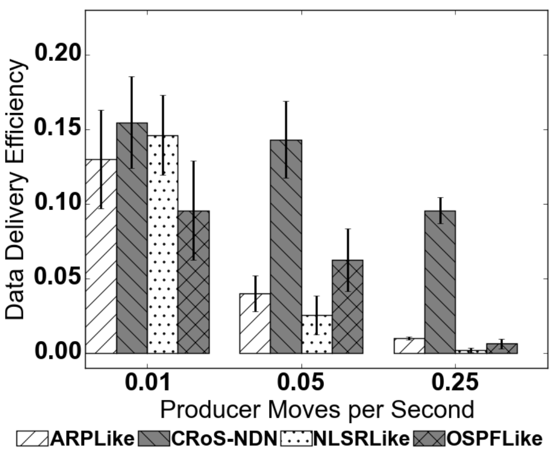
We note that one can improve NLSRLike employing direct flooding of new LSAs on the network and, then, one can avoid the slow convergence of the LSDB Synchronization procedure for new LSAs. Moreover, unlike OSPFLike, NLSRLike avoids the



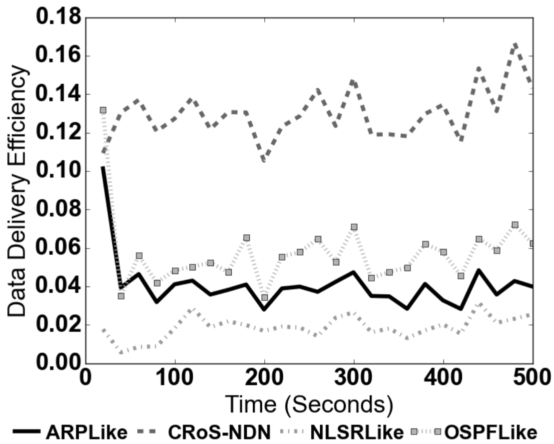
(a) 1 announced prefix.



(b) Producer rate of moves 0.05. Max and mean error: 0.054 and 0.015.

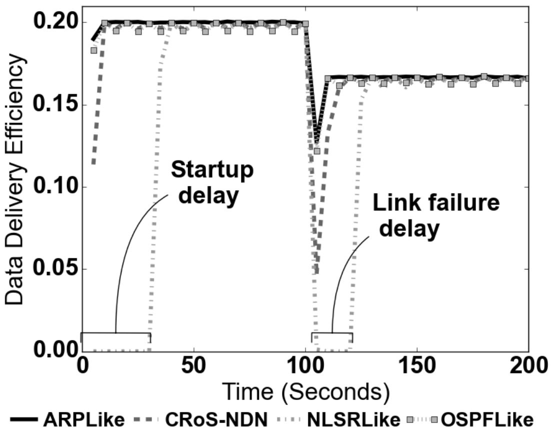


(c) 10 announced prefixes.

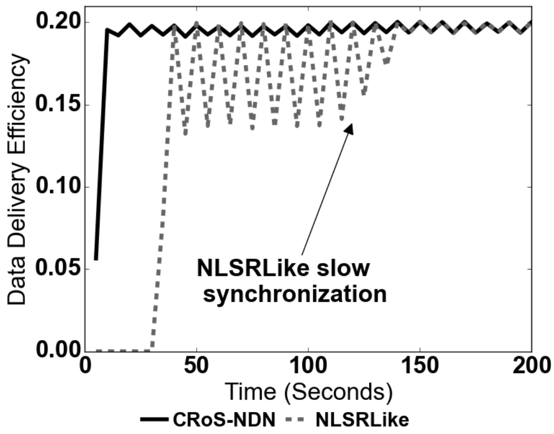


(d) Producer rate of moves 0.05. Max and mean error: 0.035 and 0.016.

Figure 5.17: Data delivery efficiency decrease due to the increase of the producer mobility and the number of named-data consumed prefixes: 1 prefix (figures a and b) and 10 prefixes (figures c and d).



(a) Start up and recovery to secondary path. Max and mean error: 0.032 and 0.001.



(b) Registration of producer new prefixes. Max and mean error: 0.006 and 0.001.

Figure 5.18: a) Data delivery delay (DDD) inference from the efficiency convergence latency at start up and after a link failure. b) CRoS-NDN and NLSRLike convergence delay for a producer registering 100 new prefixes at rate of 1 register per second.

need to recurrently flood content prefixes because NLSRLike routers synchronizes theirs local LSDB databases and, therefore, NLSRLike avoids the OSPFLike efficiency decrease with the number of prefixes. However, it is of utmost importance to observe that each NLSRLike router stores locally all network adjacency and all the content localization. Therefore, the number of routers and the number of contents impose serious scalability limitations on the amount of storage and processing power that each NLSRLike router must individually support. On the other hand, our proposal CRoS-NDN consolidates these resources on the controller function that can be executed by multiple nodes hosted in well-connected data center infrastructure and, then, CRoS-NDN routers focus the data plane functions in momently consumed content.

Chapter 6

Conclusion

We proposed and analyzed the performance of the Controller-based Routing Scheme for Named-Data Networking (CRoS-NDN). Our proposal employs the same interest and data packets defined by Named-Data Networking (NDN), and, therefore, preserves the original NDN features. CRoS-NDN is composed of two phases: the Bootstrap phase, which monitors the nodes and assures the knowledge of the global network topology, and the Named-Data Routing phase, which assures the localization and access to the requested content. The proposal fits well for data-center based network infrastructure that consolidates the network vision and offers the required storage and processing resources. The controller stores the content locations, calculates routes from the consumer to the producer and its network global view helps to avoid unnecessary message overhead, providing an efficient data delivery with low delay. Moreover, our scheme splits content names from content localization and, thus, content can be consumed from any location. This content placement freedom brings the known benefits of peer-to-peer and CDN networks that place content copies closer to consumers and, additionally, it improves content mobility efficiency.

We define specific names and specific procedures for routers and controller efficient communication over NDN. Therefore, CRoS-NDN preserves the named-data packet-forwarding scheme of NDN. In other words, unlike OpenFlow-based solutions, our proposal removes the dependency on IP for a consolidated control plane.

The proposed scheme automates the configuration of routers and controller and, thus, it avoids manual provisioning of network routers. The autonomous discovery/configuration operates correctly at start up, after topology changes, and on data mobility. Additionally, CRoS-NDN reduces the routers-controller communication overhead by i) coding routing information on content names, ii) reactively updating the controller upon routers local information change, iii) avoiding the replications of routing information from controller to routers, iv) installing a new end-to-end route on all routers in consumer-producer path with a single route request to controller, and v) restricting the interest flooding.

The efficient procedure for communication of routers and controller removes invalid routes upon no-response interest time-expiration and requests new routes to controller. Additionally, the route requester router instructs the content path routers to add the new route. Furthermore, the controller only receives interests for the update of control plane upon topology changes or data mobility. Therefore, the scheme efficiency is close to optimum for scenarios with sufficient FIB memory for simultaneous consumed prefixes and with consumer interest rate higher than the hello rate, the rate of topology changes, and the rate of data mobility. In addition, CRoS-NDN shows robust efficiency when the topology grows and the consumer interest rate increases proportionally, thus, the scheme scales well for a controller with enough resources.

CRoS-NDN reduces the routers FIB memory requirement by storing only the lastly added forwarding rules. Furthermore, the scheme correctly operates under insufficient FIB memory for simultaneous consumed prefixes, but the data delivery efficiency decreases proportionally to the rate of route requests to controller. Moreover, the scheme ensures a valid working path from consumer to producer, but it does not assure the shortest path.

The scheme shows stable convergence delay after network changes for different Hello interest rates. This is a consequence of the route-install procedure that detects the connectivity failure in consumer to producer path. Furthermore, the consumer to producer distance has a major influence on the data delivery efficiency compared to other parameters.

We derived lower bound analytical expressions for the data delivery efficiency and upper bound ones for the data delivery delay of our proposal and other known routing schemes. We employed the obtained expressions to discuss the limitation factors of each scheme. Furthermore, we evaluate and compare these schemes with simulations to validate and extend the analytical analysis.

The analytical analysis and the simulation results show that CRoS-NDN has the best performance for a set of scenarios and more robust performance over a wider range of scenarios, while the other schemes only show a high efficiency for limited ranges. CRoS-NDN shows a stable efficiency with the number of available prefixes while OSPFLike efficiency quickly decreases. CRoS-NDN shows a more robust efficiency with the limitation of FIB memory while ARPLike efficiency abruptly decreases. CRoS-NDN shows a better efficiency with producer mobility while the other schemes show a stronger efficiency reduction. CRoS-NDN shows a competitive delay in comparison with ARPLike and OSPFlike. Furthermore, we show that NLSRLike has a higher convergence delay besides the higher amount of resources in each router.

The simulation results show that CRoS-NDN is resilient to link failure recovery. It presents a fast convergence because of the rapid exchange of update messages between the controller and routers signaling the fault and updating new routes, while the NLSRLike convergence is accomplished in hop-by-hop fashion. In addition, CRoS-NDN shows an economical use of computational resources for a growing number of prefixes.

The evaluation demonstrates that CRoS-NDN registration of content copies location improves the efficiency over the cache only along the path approach. The analysis shows the lower is the cache size at routers, the higher is the efficiency gain due to the registration of copies at consumers. Additionally, the evaluation validates the results in a set of topologies with different sizes and diameters.

The results point that CRoS-NDN has a potential bottleneck at the access link to controller when content requests show prefix popularity distribution with long tail and the core routers FIB memory is insufficient for the most of momently requested prefixes. We identified a potential solution combining our CRoS-NDN scheme with a tunnelling approach. We evaluated the combined solution, called CRoS-NDN Tunnel Extension. The results show that the CRoS-NDN Tunnel Extension reduces the route requests to controller under FIB memory restriction, but the extension increases the cache misses at routers due to distinct names referring to the same content. Both the route requests and the cache misses reduce the data delivery efficiency. This result indicates that the FIB memory size must be designed in accordance to the size of router caches in order to maximize the efficiency and the use of storage resources.

For future work, we envision to further explore the tunnel solution and to evaluate the Afanasyev *et al.* proposal [37] combined with our CRoS-NDN scheme. Afanasyev *et al.* solution does not change content names, but it requires a modification in PIT structure to store the prefix-match employed in interest forwarding. The prefix-math recording requirement arises from CRoS-NDN FIB entry removal upon PIT entry expiration. We also suggest to explore our scheme in Future Internet Testbed with Security (FITS) [43] employing CCNx [78] and NFD [79] distributions with multi-controller taking care of domains.

Bibliography

[1] POSTEL, J. *Internet Protocol*. RFC 791, RFC Editor, September 1981. Available: <https://tools.ietf.org/html/rfc791>.

[2] CISCO. "Cisco Visual Networking Index: Forecast and Methodology, 2014-2019. White Paper." may 2015.

[3] CISCO. "The Zettabyte Era—Trends and Analysis. White Paper." jun 2015.

[4] ZHANG, L., AFANASYEV, A., BURKE, J., et al. "Named Data Networking", *SIGCOMM Comput. Commun. Rev.*, v. 44, n. 3, pp. 66–73, jul. 2014. ISSN: 0146-4833. Available: <http://doi.acm.org/10.1145/2656877.2656887>.

[5] YI, C., AFANASYEV, A., WANG, L., et al. "Adaptive Forwarding in Named Data Networking", *SIGCOMM Comput. Commun. Rev.*, v. 42, n. 3, pp. 62–67, jun. 2012. ISSN: 0146-4833. Available: <http://doi.acm.org/10.1145/2317307.2317319>.

[6] YI, C., AFANASYEV, A., MOISEENKO, I., et al. "A Case for Stateful Forwarding Plane", *Comput. Commun.*, v. 36, n. 7, pp. 779–791, abr. 2013. ISSN: 0140-3664. Available: <http://dx.doi.org/10.1016/j.comcom.2013.01.005>.

[7] CAROFIGLIO, G., GEHLEN, V., PERINO, D. "Experimental Evaluation of Memory Management in Content-Centric Networking". In: *Communications (ICC), 2011 IEEE International Conference on*, pp. 1–6, June 2011. Available: <http://dx.doi.org/10.1109/icc.2011.5962739>.

[8] JACOBSON, V., SMETTERS, D. K., THORNTON, J. D., et al. "Networking Named Content". In: *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '09, pp. 1–12, New York, NY, USA, 2009. ACM. ISBN: 978-1-60558-636-6. Available: <http://doi.acm.org/10.1145/1658939.1658941>.

[9] AFANASYEV, A., MAHADEVAN, P., MOISEENKO, I., et al. "Interest flooding attack and countermeasures in Named Data Networking". In: *IFIP Networking Conference, 2013*, pp. 1–9, May 2013. Available: <http://named-data.net/wp-content/uploads/ifip-interest-flooding-ndn.pdf>.

[10] PERINO, D., VARVELLO, M. "A Reality Check for Content Centric Networking". In: *Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking*, ICN '11, pp. 44–49, New York, NY, USA, 2011. ACM. ISBN: 978-1-4503-0801-4. Available: <http://doi.acm.org/10.1145/2018584.2018596>.

[11] CHOI, H.-G., YOO, J., CHUNG, T., et al. "CoRC: Coordinated Routing and Caching for Named Data Networking". In: *Proceedings of the Tenth ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, ANCS '14, pp. 161–172, New York, NY, USA, 2014. ACM. ISBN: 978-1-4503-2839-5. Available: <http://doi.acm.org/10.1145/2658260.2658275>.

[12] FAYAZBAKHSH, S. K., LIN, Y., TOOTOONCHIAN, A., et al. "Less Pain, Most of the Gain: Incrementally Deployable ICN", *SIGCOMM Comput. Commun. Rev.*, v. 43, n. 4, pp. 147–158, ago. 2013. ISSN: 0146-4833. Available: <http://doi.acm.org/10.1145/2534169.2486023>.

[13] GHODSI, A., SHENKER, S., KOPONEN, T., et al. "Information-centric Networking: Seeing the Forest for the trees". In: *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, HotNets-X, pp. 1:1–1:6. ACM, 2011. Available: <http://dx.doi.org/10.1145/2070562.2070563>.

[14] DETTI, A., POMPOSINI, M., BLEFARI-MELAZZI, N., et al. "Supporting the Web with an Information Centric Network That Routes by Name", *Comput. Netw.*, v. 56, n. 17, pp. 3705–3722, nov. 2012. ISSN: 1389-1286. Available: <http://dx.doi.org/10.1016/j.comnet.2012.08.006>.

[15] IMBRENDA, C., MUSCARIELLO, L., ROSSI, D. "Analyzing Cacheable Traffic in Isp Access Networks for Micro Cdn Applications via Content-centric Networking". In: *Proceedings of the 1st International Conference on Information-centric Networking*, ICN '14, pp. 57–66, New York, NY, USA, 2014. ACM. ISBN: 978-1-4503-3206-4. Available: <http://doi.acm.org/10.1145/2660129.2660146>.

[16] ROSSINI, G., ROSSI, D. "Coupling Caching and Forwarding: Benefits, Analysis, and Implementation". In: *Proceedings of the 1st International Con-*

ference on Information-centric Networking, ICN '14, pp. 127–136, New York, NY, USA, 2014. ACM. ISBN: 978-1-4503-3206-4. Available: <http://doi.acm.org/10.1145/2660129.2660153>.

[17] SOURLAS, V., FLEGKAS, P., TASSIULAS, L. "A Novel Cache Aware Routing Scheme for Information-Centric Networks", *Comput. Netw.*, v. 59, pp. 44–61, fev. 2014. ISSN: 1389-1286. Available: <http://dx.doi.org/10.1016/j.bjp.2013.12.002>.

[18] FENG, B., ZHOU, H., GAO, S., et al. "An Exploration of Cache Collaboration in Information-centric Network", *Int. J. Commun. Syst.*, v. 27, n. 9, pp. 1243–1267, set. 2014. ISSN: 1074-5351. Available: <http://dx.doi.org/10.1002/dac.2773>.

[19] DABIRMOGHADDAM, A., BARIJOUGH, M. M., GARCIA-LUNA-ACEVES, J. "Understanding Optimal Caching and Opportunistic Caching at "the Edge" of Information-centric Networks". In: *Proceedings of the 1st International Conference on Information-centric Networking*, ICN '14, pp. 47–56, New York, NY, USA, 2014. ACM. ISBN: 978-1-4503-3206-4. Available: <http://doi.acm.org/10.1145/2660129.2660143>.

[20] BARI, M., CHOWDHURY, S., AHMED, R., et al. "A survey of naming and routing in information-centric networks", *Communications Magazine, IEEE*, v. 50, n. 12, pp. 44–53, December 2012. ISSN: 0163-6804. Available: <http://dx.doi.org/10.1109/MCOM.2012.6384450>.

[21] AHLGREN, B., DANNEWITZ, C., IMBRENDA, C., et al. "A survey of information-centric networking", *Communications Magazine, IEEE*, v. 50, n. 7, pp. 26–36, July 2012. ISSN: 0163-6804. Available: <http://dx.doi.org/10.1109/MCOM.2012.6231276>.

[22] XYLOMENOS, G., VERVERIDIS, C., SIRIS, V., et al. "A Survey of Information-Centric Networking Research", *Communications Surveys Tutorials, IEEE*, v. 16, n. 2, pp. 1024–1049, Second 2014. ISSN: 1553-877X. Available: <http://dx.doi.org/10.1109/SURV.2013.070813.00063>.

[23] ZHANG, M., LUO, H., ZHANG, H. "A Survey of Caching Mechanisms in Information-Centric Networking", *IEEE Communications Surveys Tutorials*, v. 17, n. 3, pp. 1473–1499, thirdquarter 2015. ISSN: 1553-877X. Available: <http://dx.doi.org/10.1109/COMST.2015.2420097>.

[24] FANG, C., YU, F. R., HUANG, T., et al. "A Survey of Green Information-Centric Networking: Research Issues and Challenges", *IEEE Communi-*

cations Surveys Tutorials, v. 17, n. 3, pp. 1455–1472, thirdquarter 2015. ISSN: 1553-877X. Available: <`http://dx.doi.org/10.1109/COMST.2015.2394307`>.

[25] SONIYA, M. M. S., KUMAR, K. "A survey on named data networking". In: *Electronics and Communication Systems (ICECS), 2015 2nd International Conference on*, pp. 1515–1519, Feb 2015. Available: <`http://dx.doi.org/10.1109/ECS.2015.7124841`>.

[26] CAROFIGLIO, G., MORABITO, G., MUSCARIELLO, L., et al. "From Content Delivery Today to Information Centric Networking", *Comput. Netw.*, v. 57, n. 16, pp. 3116–3127, nov. 2013. ISSN: 1389-1286. Available: <`http://dx.doi.org/10.1016/j.comnet.2013.07.002`>.

[27] CAROFIGLIO, G., GALLO, M., MUSCARIELLO, L. "On the Performance of Bandwidth and Storage Sharing in Information-centric Networks", *Comput. Netw.*, v. 57, n. 17, pp. 3743–3758, dez. 2013. ISSN: 1389-1286. Available: <`http://dx.doi.org/10.1016/j.comnet.2013.08.018`>.

[28] AMADEO, M., CAMPOLO, C., QUEVEDO, J., et al. "Information-centric networking for the internet of things: challenges and opportunities", *IEEE Network*, v. 30, n. 2, pp. 92–100, March 2016. ISSN: 0890-8044. Available: <`http://dx.doi.org/10.1109/MNET.2016.7437030`>.

[29] TYSON, G., SASTRY, N., CUEVAS, R., et al. "A Survey of Mobility in Information-centric Networks", *Commun. ACM*, v. 56, n. 12, pp. 90–98, dez. 2013. ISSN: 0001-0782. Available: <`http://doi.acm.org/10.1145/2500501`>.

[30] KUROSE, J. "Information-centric networking: The evolution from circuits to packets to content", *Computer Networks*, v. 66, pp. 112–120, 2014. Available: <`http://dx.doi.org/10.1016/j.comnet.2014.04.002`>.

[31] TROSSEN, D., SATHIASEELAN, A., OTT, J. "Towards an Information Centric Network Architecture for Universal Internet Access", *SIGCOMM Comput. Commun. Rev.*, v. 46, n. 1, pp. 44–49, jan. 2016. ISSN: 0146-4833. Available: <`http://doi.acm.org/10.1145/2875951.2875959`>.

[32] KOPONEN, T., CHAWLA, M., CHUN, B.-G., et al. "A Data-oriented (and Beyond) Network Architecture", *SIGCOMM Comput. Commun. Rev.*, v. 37, n. 4, pp. 181–192, ago. 2007. ISSN: 0146-4833. Available: <`http://doi.acm.org/10.1145/1282427.1282402`>.

[33] JOKELA, P., ZAHEMSZKY, A., ESTEVE ROTHENBERG, C., et al. "LIPSIN: Line Speed Publish/Subscribe Inter-networking", *SIGCOMM Comput. Commun. Rev.*, v. 39, n. 4, pp. 195–206, ago. 2009. ISSN: 0146-4833. Available: <`http://doi.acm.org/10.1145/1594977.1592592`>.

[34] CARZANIGA, A., PAPALINI, M., WOLF, A. L. "Content-based Publish/Subscribe Networking and Information-centric Networking". In: *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, ICN '11, pp. 56–61. ACM, 2011. Available: <`http://doi.acm.org/10.1145/2018584.2018599`>.

[35] AHMED, R., BARI, F., CHOWDHURY, S. R., et al. "Alpha Route: Routing on Names", *IEEE/ACM Transactions on Networking*, v. PP, n. 99, pp. 1–1, 2016. ISSN: 1063-6692. Available: <`http://dx.doi.org/10.1109/TNET.2015.2506617`>.

[36] LOUATI, W., BEN-AMEUR, W., ZEGHLACHE, D. "A bottleneck-free tree-based name resolution system for Information-Centric Networking", *Computer Networks*, v. 91, pp. 341 – 355, 2015. ISSN: 1389-1286. Available: <`http://dx.doi.org/10.1016/j.comnet.2015.08.024`>.

[37] AFANASYEV, A., YI, C., WANG, L., et al. "SNAMP: Secure namespace mapping to scale NDN forwarding". In: *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on*, pp. 281–286, April 2015. Available: <`http://dx.doi.org/10.1109/INFCOMW.2015.7179398`>.

[38] BAID, A., VU, T., RAYCHAUDHURI, D. "Comparing alternative approaches for networking of named objects in the future Internet". In: *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pp. 298–303, March 2012. Available: <`http://dx.doi.org/10.1109/INFCOMW.2012.6193509`>.

[39] DANNEWITZ, C., D'AMBROSIO, M., VERCELLONE, V. "Hierarchical DHT-based Name Resolution for Information-centric Networks", *Comput. Commun.*, v. 36, n. 7, pp. 736–749, abr. 2013. ISSN: 0140-3664. Available: <`http://dx.doi.org/10.1016/j.comcom.2013.01.014`>.

[40] CAESAR, M., CONDIE, T., KANNAN, J., et al. "ROFL: Routing on Flat Labels", *SIGCOMM Comput. Commun. Rev.*, v. 36, n. 4, pp. 363–374, ago. 2006. ISSN: 0146-4833. Available: <`http://doi.acm.org/10.1145/1151659.1159955`>.

[41] MCKEOWN, N., ANDERSON, T., BALAKRISHNAN, H., et al. "OpenFlow: Enabling Innovation in Campus Networks", *ACM SIGCOMM Computer Communication Review*, v. 38, n. 2, pp. 69–74, April 2008. Available: <http://dx.doi.org/10.1145/1355734.1355746>.

[42] MATTOS, D., FERNANDES, N., DA COSTA, V., et al. "OMNI: OpenFlow MaNagement Infrastructure". In: *Network of the Future (NOF), 2011 International Conference on the*, pp. 52 –56, nov. 2011. Available: <http://dx.doi.org/10.1109/NOF.2011.6126682>.

[43] FITS. "Future Internet Testbed with Security". 2016. Available: <http://www.gta.ufrj.br/fits/>.

[44] KREUTZ, D., RAMOS, F. M. V., VERÍSSIMO, P. E., et al. "Software-Defined Networking: A Comprehensive Survey", *Proceedings of the IEEE*, v. 103, n. 1, pp. 14–76, Jan 2015. ISSN: 0018-9219. Available: <http://dx.doi.org/10.1109/JPROC.2014.2371999>.

[45] FERNANDES, N., MOREIRA, M., MORAES, I., et al. "Virtual Networks: Isolation, Performance, and Trends", *Annals of Telecommunications*, v. 66, n. 5-6, pp. 339–355, 2011. doi: 10.1007/s12243-010-0208-9. Available: <http://dx.doi.org/10.1007/s12243-010-0208-9>.

[46] ROTHENBERG, C. E., NASCIMENTO, M. R., SALVADOR, M. R., et al. "Revisiting routing control platforms with the eyes and muscles of software-defined networking". In: *Proceedings of the first workshop on Hot topics in software defined networks*, HotSDN '12, pp. 13–18, New York, NY, USA, 2012. ACM. ISBN: 978-1-4503-1477-0. Available: <http://doi.acm.org/10.1145/2342441.2342445>.

[47] SHI, W., AFANASYEV, A. "RepoSync: Combined action-based and data-based synchronization model in Named Data Network". In: *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on*, pp. 275–280, April 2015. Available: <http://dx.doi.org/10.1109/INFCOMW.2015.7179397>.

[48] GAO, S., ZENG, Y., LUO, H., et al. "Scalable control plane for intra-domain communication in software defined information centric networking", *Future Generation Computer Systems*, v. 56, pp. 110 – 120, 2016. ISSN: 0167-739X. Available: <http://dx.doi.org/10.1016/j.future.2015.10.017>.

[49] SALSANO, S., BLEFARI-MELAZZI, N., DETTI, A., et al. "Information Centric Networking over SDN and OpenFlow: Architectural Aspects and Experiments on the OFELIA Testbed", *Comput. Netw.*, v. 57, n. 16, pp. 3207–3221, nov. 2013. ISSN: 1389-1286. Available: <http://dx.doi.org/10.1016/j.comnet.2013.07.031>.

[50] ATKINSON, R., BHATTI, S. *Identifier-Locator Network Protocol (ILNP) Architectural Description.* RFC 6740, RFC Editor, November 2012. Available: <https://tools.ietf.org/html/rfc6740>.

[51] PSARAS, I., CHAI, W. K., PAVLOU, G. "In-Network Cache Management and Resource Allocation for Information-Centric Networks", *IEEE Transactions on Parallel and Distributed Systems*, v. 25, n. 11, pp. 2920–2931, Nov 2014. ISSN: 1045-9219. Available: <http://dx.doi.org/10.1109/TPDS.2013.304>.

[52] JACOBSON, V., SMETTERS, D. K., BRIGGS, N. H., et al. "VoCCN: Voice-over Content-centric Networks". In: *Proceedings of the 2009 Workshop on Re-architecting the Internet*, ReArch '09, pp. 1–6, New York, NY, USA, 2009. ACM. ISBN: 978-1-60558-749-3. Available: <http://doi.acm.org/10.1145/1658978.1658980>.

[53] TORRES, J. V., BOUTABA, R., DUARTE, O. C. M. B. *An Autonomous and Efficient Controller-based Routing Scheme for Networking Named-Data Mobility.* Technical report, Electrical Engineering Program, COPPE/UFRJ, April 2016. Available: <http://www.gta.ufrj.br/ftp/gta/TechReports/TBD16.pdf>.

[54] TORRES, J. V., BOUTABA, R., DUARTE, O. C. M. B. *Evaluating CRoS-NDN: A comparative performance analysis of the Controller-based Routing Scheme for Named-Data Networking.* Technical report, Electrical Engineering Program, COPPE/UFRJ, May 2016. Available: <http://www.gta.ufrj.br/ftp/gta/TechReports/TBD16a.pdf>.

[55] RULA, J. P., BUSTAMANTE, F. E. "Behind the Curtain: Cellular DNS and Content Replica Selection". In: *Proceedings of the 2014 Conference on Internet Measurement Conference*, IMC '14, pp. 59–72, New York, NY, USA, 2014. ACM. ISBN: 978-1-4503-3213-2. Available: <http://doi.acm.org/10.1145/2663716.2663734>.

[56] MIJUMBI, R., SERRAT, J., GORRICHO, J. L., et al. "Network Function Virtualization: State-of-the-Art and Research Challenges", *IEEE Communications Surveys Tutorials*, v. 18, n. 1, pp. 236–262, Firstquarter

2016. ISSN: 1553-877X. Available: <http://dx.doi.org/10.1109/COMST.2015.2477041>.

[57] AFANASYEV, A., MOISEENKO, I., ZHANG, L. *ndnSIM: NDN simulator for NS-3*. Technical report, University of California, Los Angeles, 2012. Available: <http://named-data.net/wp-content/uploads/TRndnsim.pdf>.

[58] MUDIGONDA, J., YALAGANDULA, P., AL-FARES, M., et al. "SPAIN: COTS data-center Ethernet for multipathing over arbitrary topologies". In: *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, NSDI'10, pp. 18–18, 2010. Available: <http://dl.acm.org/citation.cfm?id=1855711.1855729>.

[59] COMPAGNO, A., CONTI, M., GHALI, C., et al. "To NACK or Not to NACK? Negative Acknowledgments in Information-Centric Networking". In: *Computer Communication and Networks (ICCCN), 2015 24th International Conference on*, pp. 1–10, Aug 2015. Available: <http://dx.doi.org/10.1109/ICCCN.2015.7288477>.

[60] CAROFIGLIO, G., GALLO, M., MUSCARIELLO, L., et al. "Pending Interest Table Sizing in Named Data Networking". In: *Proceedings of the 2Nd International Conference on Information-Centric Networking*, ICN '15, pp. 49–58, New York, NY, USA, 2015. ACM. ISBN: 978-1-4503-3855-4. Available: <http://doi.acm.org/10.1145/2810156.2810167>.

[61] VIRGILIO, M., MARCHETTO, G., SISTO, R. "PIT Overload Analysis in Content Centric Networks". In: *Proceedings of the 3rd ACM SIGCOMM Workshop on Information-centric Networking*, ICN '13, pp. 67–72, New York, NY, USA, 2013. ACM. ISBN: 978-1-4503-2179-2. Available: <http://doi.acm.org/10.1145/2491224.2491225>.

[62] DAI, H., LIU, B., CHEN, Y., et al. "On Pending Interest Table in Named Data Networking". In: *Proceedings of the Eighth ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, ANCS '12, pp. 211–222, New York, NY, USA, 2012. ACM. ISBN: 978-1-4503-1685-9. Available: <http://doi.acm.org/10.1145/2396556.2396600>.

[63] GARCIA-LUNA-ACEVES, J. J., MIRZAZAD-BARIJOUGH, M. "Enabling correct interest forwarding and retransmissions in a content centric network". In: *Architectures for Networking and Communications Systems (ANCS), 2015 ACM/IEEE Symposium on*, pp. 135–146, May 2015. Available: <http://dx.doi.org/10.1109/ANCS.2015.7110127>.

[64] CHIOCCHETTI, R., PERINO, D., CAROFIGLIO, G., et al. "INFORM: A Dynamic Interest Forwarding Mechanism for Information Centric Networking". In: *Proceedings of the 3rd ACM SIGCOMM Workshop on Information-centric Networking*, ICN '13, pp. 9–14, New York, NY, USA, 2013. ACM. ISBN: 978-1-4503-2179-2. Available: <`http://doi.acm.org/10.1145/2491224.2491227`>.

[65] CHIOCCHETTI, R., ROSSI, D., ROSSINI, G., et al. "Exploit the Known or Explore the Unknown?: Hamlet-like Doubts in ICN". In: *Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking*, ICN '12, pp. 7–12, New York, NY, USA, 2012. ACM. ISBN: 978-1-4503-1479-4. Available: <`http://doi.acm.org/10.1145/2342488.2342491`>.

[66] WANG, L., BAYHAN, S., OTT, J., et al. "Pro-Diluvian: Understanding Scoped-Flooding for Content Discovery in Information-Centric Networking". In: *Proceedings of the 2Nd International Conference on Information-Centric Networking*, ICN '15, pp. 9–18, New York, NY, USA, 2015. ACM. ISBN: 978-1-4503-3855-4. Available: <`http://doi.acm.org/10.1145/2810156.2810162`>.

[67] PLUMMER, D. C. *An Ethernet Address Resolution Protocol – or – Converting Network Protocol Addresses*. RFC 826, RFC Editor, November 1982. Available: <`https://tools.ietf.org/html/rfc826`>.

[68] HEDRICK, C. *Routing Information Protocol*. RFC 1058, RFC Editor, June 1988. Available: <`https://tools.ietf.org/html/rfc1058`>.

[69] MOY, J. *OSPF Version 2*. RFC 2328, RFC Editor, April 1998. Available: <`https://tools.ietf.org/html/rfc2328`>.

[70] HOQUE, A. K. M. M., AMIN, S. O., ALYYAN, A., et al. "NLSR: Named-data Link State Routing Protocol". In: *Proceedings of the 3rd ACM SIGCOMM Workshop on Information-centric Networking*, ICN '13, pp. 15–20, New York, NY, USA, 2013. ACM. ISBN: 978-1-4503-2179-2. Available: <`http://doi.acm.org/10.1145/2491224.2491231`>.

[71] HEMMATI, E., GARCIA-LUNA-ACEVES, J. "A New Approach to Name-Based Link-State Routing for Information-Centric Networks". In: *Proceedings of the 2Nd International Conference on Information-Centric Networking*, ICN '15, pp. 29–38, New York, NY, USA, 2015. ACM. ISBN: 978-1-4503-3855-4. Available: <`http://doi.acm.org/10.1145/2810156.2810173`>.

[72] WANG, L., KANGASHARJU, J., SATHIASEELAN, A., et al. "You Really Need A Good Ruler to Measure Caching Performance in Information-Centric Networks", *CoRR*, v. abs/1603.05963, 2016. Available: <http://arxiv.org/abs/1603.05963>.

[73] WANG, Y., LI, Z., TYSON, G., et al. "Design and Evaluation of the Optimal Cache Allocation for Content-Centric Networking", *IEEE Transactions on Computers*, v. 65, n. 1, pp. 95–107, Jan 2016. ISSN: 0018-9340. Available: <http://dx.doi.org/10.1109/TC.2015.2409848>.

[74] CHAI, W. K., HE, D., PSARAS, I., et al. "Cache "Less for More" in Information-centric Networks (Extended Version)", *Comput. Commun.*, v. 36, n. 7, pp. 758–770, abr. 2013. ISSN: 0140-3664. Available: <http://dx.doi.org/10.1016/j.comcom.2013.01.007>.

[75] SPRING, N., MAHAJAN, R., WETHERALL, D., et al. "Measuring ISP Topologies with Rocketfuel", *IEEE/ACM Trans. Netw.*, v. 12, n. 1, pp. 2–16, fev. 2004. ISSN: 1063-6692. Available: <http://dx.doi.org/10.1109/TNET.2003.822655>.

[76] ZHANG, Y., AFANASYEV, A., ZHANG, L. "A Survey of Mobility Support in Named Data Networking", *Proceedings of the third Workshop on Name-Oriented Mobility: Architecture, Algorithms and Applications (NOM'2016)*, April 2016. Available: <http://named-data.net/wp-content/uploads/2016/04/survey_mobility_support_ndn.pdf>.

[77] ZHANG, G., CHEN, X., LU, Y., et al. "Enabling Tunneling in CCN", *IEEE Communications Letters*, v. 20, n. 1, pp. 149–152, Jan 2016. ISSN: 1089-7798. Available: <http://dx.doi.org/10.1109/LCOMM.2015.2500899>.

[78] CCNX. "CCNx Project". 2016. Available: <http://www.ccnx.org/>.

[79] NFD. "Named-Data Networking Forwarding Daemon (NFD)". 2016. Available: <http://named-data.net/doc/NFD/current/>.

[80] FILSFILS, C., NAINAR, N. K., PIGNATARO, C., et al. "The Segment Routing Architecture". In: *2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec 2015. Available: <http://dx.doi.org/10.1109/GLOCOM.2015.7417124>.

Appendix A

CRoS-NDN Tunnel Extension

In chapter 5, we identified that the rate of route requests increases when there is insufficient FIB memory for most of the solicited prefixes and the prefix popularity distribution at core routers present a long tail shape. In this scenario, the controller access link congests and causes interests retransmissions. The additional interests further reduce the efficiency. In this chapter, we present the CRoS-NDN Tunnel Extension that combines our CRoS-NDN scheme and the NDN tunneling approach described in [77]. The CRoS-NDN Tunnel Extension is similar to the Segment Routing technique [80], which employs source routing and tunnelling. The tunnelling approach reduces the FIB memory requirement at core routers to the amount of prefixes employed to identify the destination network segments. In this thesis, we present only the special case that each segment identifies a single router.

The CRoS-NDN Tunnel Extension modifies the Route Installation procedure. In the Route Request procedure, the controller informs the whole route from the source router to the destination, which includes the information of the route-installation interest name to be generated by the source router. When the source router receives the route-reply data packet from the controller, the router searches the next and the last hop information in the packet content. Next, the router adds a FIB entry with prefix `/tunnel/lastHop` via the same interface it reaches the next hop router, where `lastHop` is the last hop router in the path. In addition, the route-requesting router prepend the prefix `/tunnel/lastHop` to subsequent interests with prefix name `/my-prefix` before forwarding them, where `myprefix` is the prefix that is registered in controller for the requested content name.

In the sequence, the source router creates a route-install interest packet with name `/router/hopID1/installRouteAndForward/hopID1/.../hopIDn/pre-fixSize/tunnel/lastHop/wantedcontent`, where `hopID1/.../hopIDn` is the routers sequence in the path to the prefix producer and, the `prefixSize` indicates the size of content prefix registered by the producer. The next hop router replies the interest packet and employs the sequence to create the FIB entry to the prefix `/tunnel/lastHop`. This procedure is repeated until the interest packet reaches the destination router, which already contains a FIB entry to `/myprefix` and the route is fully installed. Each router in the path adds a PIT entry to `/tunnel/lastHop/wantedcontent` and, the last hop router sends an interest to `/wantedcontent` that is replied back by the producer. In the sequence, the last hop router receives the `/wantedcontent` data and forwards to consumer the data packet for interest name `/tunnel/lastHop/wantedcontent` [1]. The first hop router removes the prefix `/tunnel/lastHop` from received data packets and forwards the data packet `/wantedcontent` to consumer. Therefore, one single interest installs the new route and requests the content. The last hop router removes the prefix `/tunnel/lastHop` from interests with name `/tunnel/lastHop/wantedcontent` and forward the interest `/wantedcontent`. Figure A.1 presents the interest and data sequence of the CRoS-NDN Tunnel Extension. Interests 8*c* and 10 show the modified content names used in the modified Route Installation procedure.

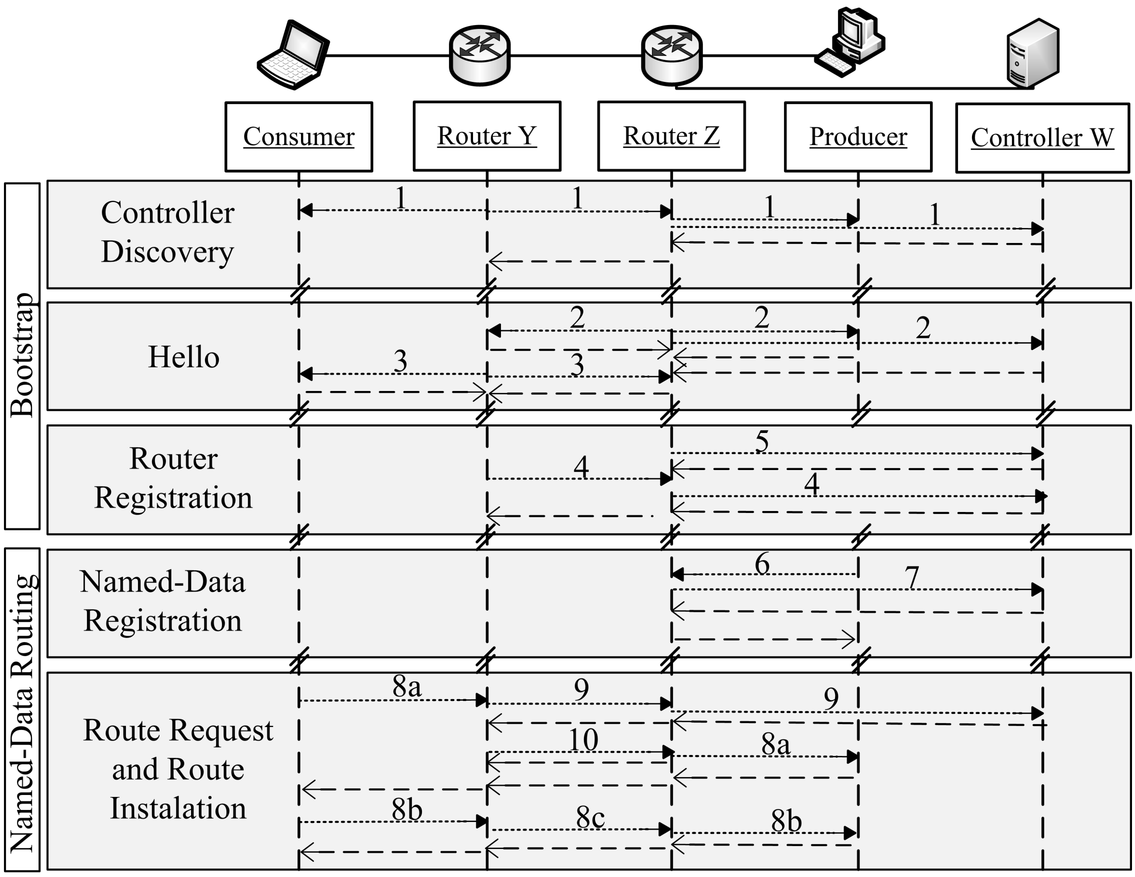
In order to evaluate the CRoS-NDN Tunnel Extension behavior in the scenario with insufficient FIB memory for the requested prefixes, we repeated the simulations with consumer interests following the Zipf-Mandelbrot distribution for content prefix popularity and a growing number of consumers. Figure A.3a shows the comparative results for the original CRoS-NDN scheme and the CRoS-NDN Tunnel Extension. The results show that the CRoS-NDN Tunnel Extension reduces the route request growth in function of the number of consumers and reduces the corresponding decrease of the data delivery efficiency. Figure A.3b show the data delivery efficiency for CRoS-NDN Tunnel Extension higher than 0.08 compared to 0.02 for CRoS-NDN in Figure A.3a. In addition, Figure A.3d show the rate of route requests for CRoS-NDN Tunnel Extension lower than 70 compared to 900 for CRoS-NDN in Figure A.3c.

On the other hand, the CRoS-NDN Tunnel Extension reduces the cache hits opportunities in the path to content and, in consequence, decreases the data delivery efficiency. The tunnelling approach changes content names according to the network segment that host the content. Therefore, requests for the same content from different consumers can have the shortest path pointing to different hosts and, then, to different content names. The distinct names implies distinct content and avoids cache hits in intersecting routers of the two paths. Figure A.2 shows an example of this scenario, where *Consumer1* first requests the content, then registers a copy location at router RA, and, afterwards, *Consumer2* requests the same content [2].

In order to evaluate the reduction of the data delivery efficiency due to the change

[1]The details in data packet signature related to changes in data name are not covered in our analysis.

[2]We omitted interests to the controller.



CRoS-NDN Tunnel Extension specific content names:

1 - /controller 2 - /hello/RouterZ 3 - /hello/RouterY

4 - /controllerx/ControllerW/registerrouter/RouterY/RouterZ

5 - /controllerx/ControllerW/registerrouter/RouterZ/RouterY

6 - /registerNamedData/myprefix

7 - /controllerx/ControllerW/registerNamedData/RouterZ/myprefix

8a - /myprefix/seq1 8b - myprefix/seq2

8c - /tunnel/RouterZ/myprefix/seq2

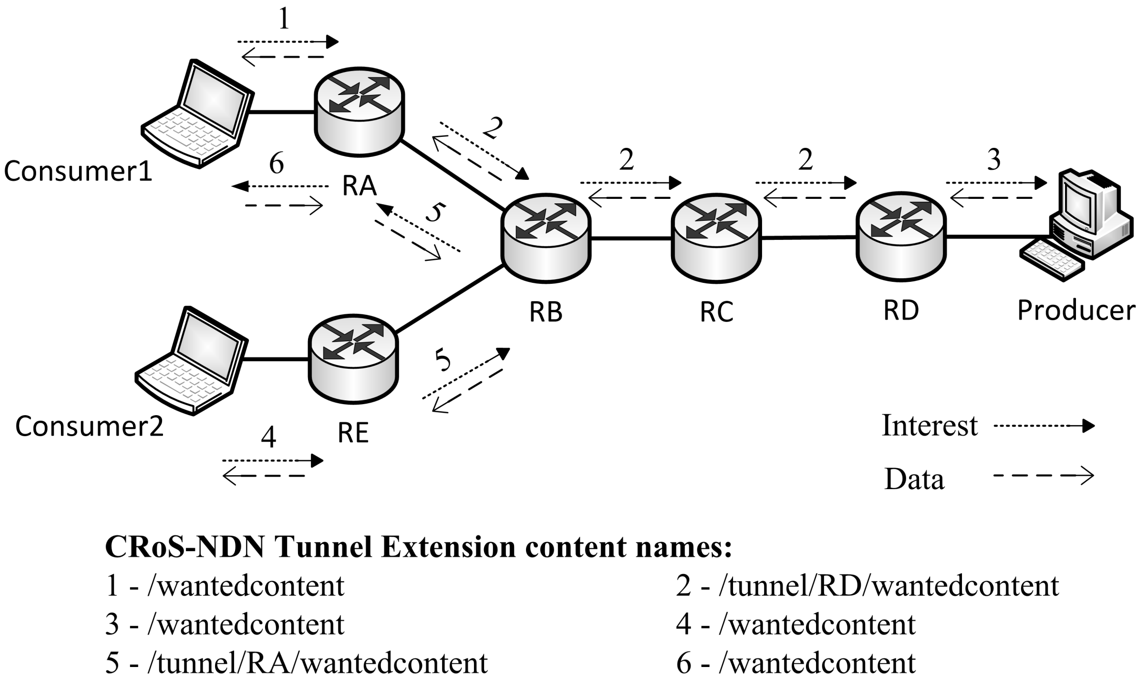
9 - /controllerx/ControllerW/routeFrom/RouterY/myprefix/seq1

10 - /router/RouterZ/installRouteAndForward/RouterZ/prefixSize/tunnel/RouterZ/myprefix/seq1

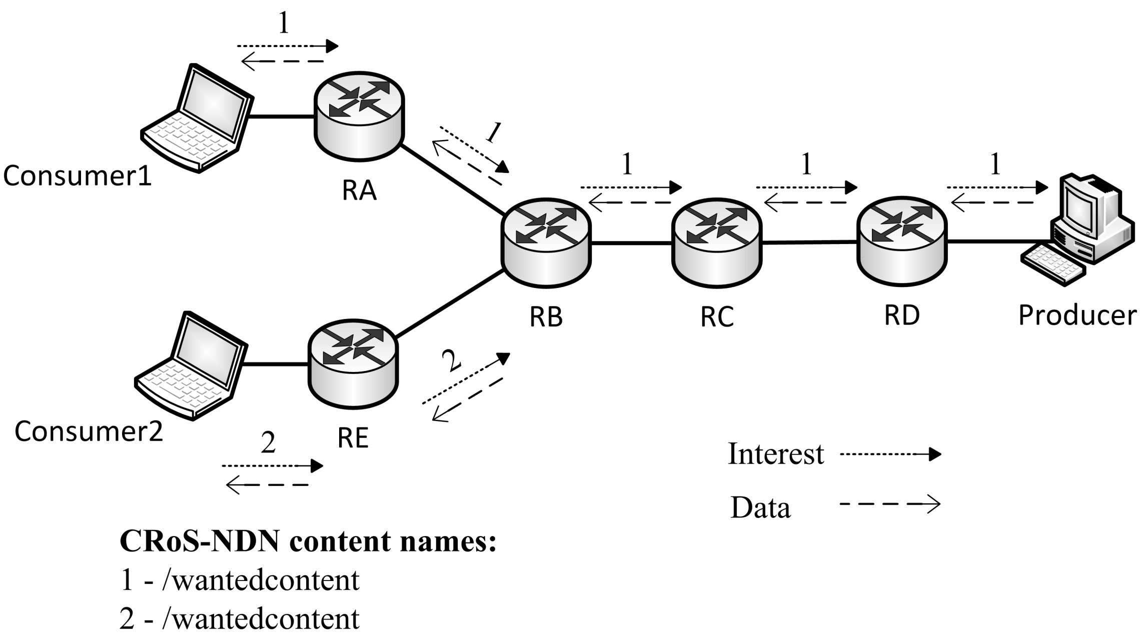
Interest

Data

Figure A.1: The Interest/Data packet time sequence for CRoS-NDN Tunnel Extension procedures. (1) Routers Y and Z find Controller W by sending a controller discovery message. (2) and (3) Routers Y and Z send a hello message to inform each other their presence. (4) and (5) Each router sends a router register message to register its neighbors in Controller W. (6) and (7) The Producer sends a named-data registration message to Router Z to register a named-data, in controller W. (8a) and (9) Consumer requests a content, sending a content-request message, and Router Y requests Controller W a new route for the named-data. (10) Router Y requests Router Z to install a new route to the named-data. (8b) Routers Y and Z forward further Consumer interests directly to Producer.



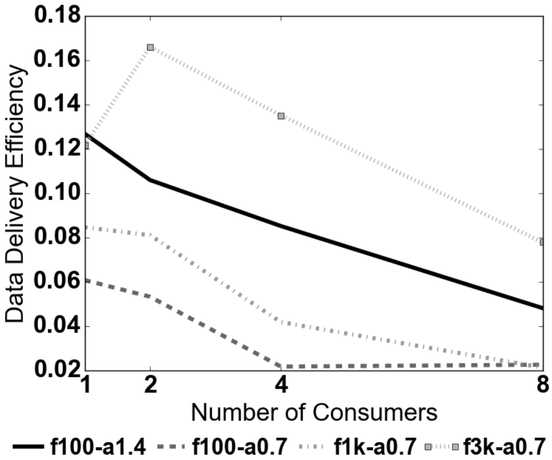
(a) Cache hit miss at router RB for CRoS-NDN Tunnel Extension due to the change of content name.

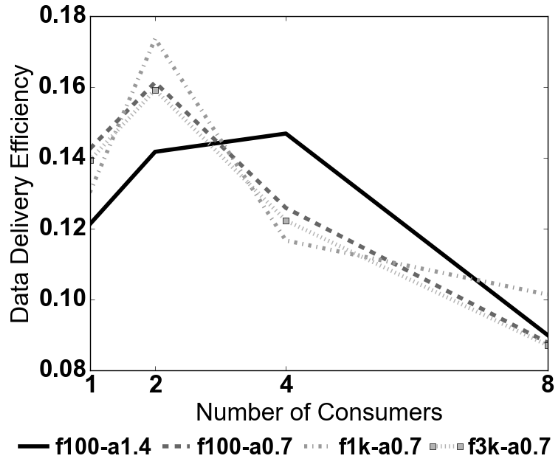


(b) Cache hit at router RB for CRoS-NDN.

Figure A.2: Cache hit miss versus cache hit for CRoS-NDN Tunnel Extension and CRoS-NDN, respectively. CRoS-NDN Tunnel Extension changes the content name and reduces the cache hit opportunities.

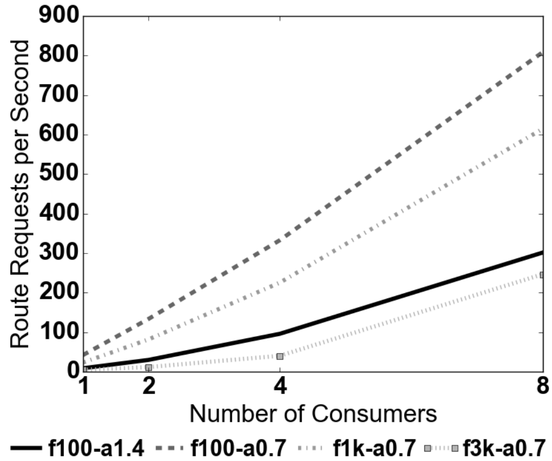
in content names, we repeated the simulations with registration of content copies. Figure A.4 shows the results for routers with cache size big enough to demonstrate the cache misses of CRoS-NDN Tunnel Extension and its lower data delivery efficiency compared to CRoS-NDN. This result indicates that the FIB memory must support the amount of prefixes corresponding to the content stored in cache of routers. In addition, the network must employ unique names for content stored in router caches for the time these content last in cache. Otherwise, either the route requests received by the controller increase due to FIB entry replacement, CRoS-NDN case, or the cache misses at routers increase due to distinct names referring to the same content, CRoS-NDN Tunnel Extension case. In both cases, the data delivery efficiency decreases from its maximum potential value.

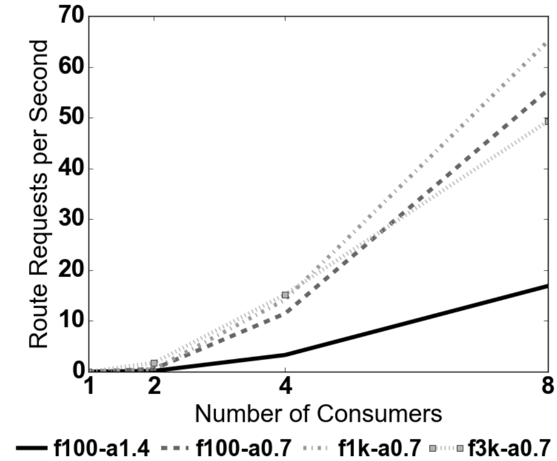




(a) CRoS-NDN. Max and mean error: 0.025 and 0.018.

(b) CRoS-NDN Tunnel Extension. Max and mean error: 0.025 and 0.020.

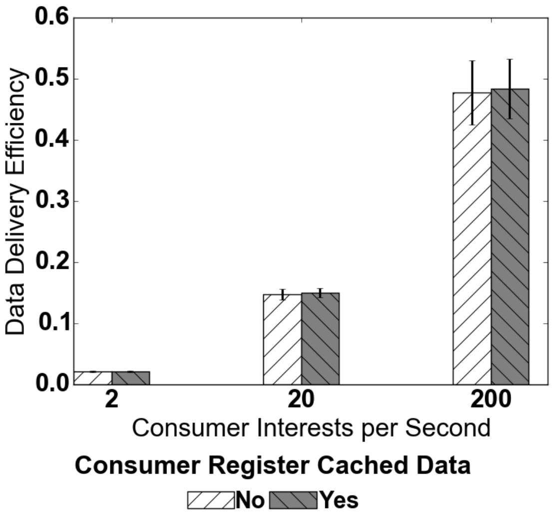




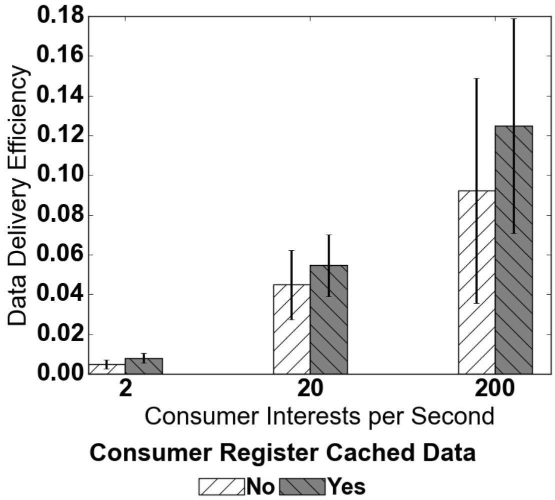
(c) Route requests received by the controller for CRoS-NDN. Max and mean error: 69 and 16.

(d) Route requests received by the controller for CRoS-NDN Tunnel Extension. Max and mean error: 13 and 4.

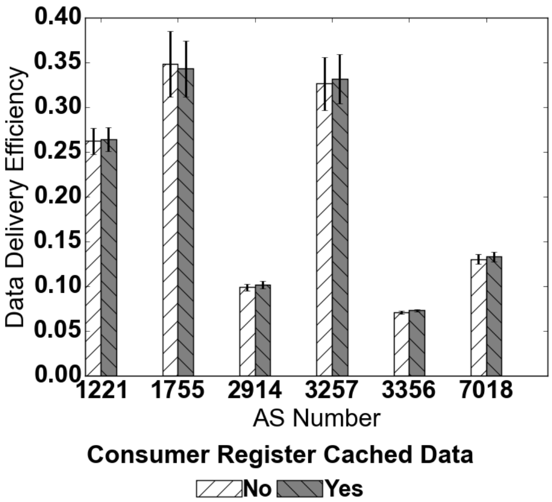
Figure A.3: Data delivery efficiency of CRoS-NDN and CRoS-NDN Tunnel Extension for consumer interests following the Zipf-Mandelbrot distribution for content prefix popularity. The simulation employs 3000 prefixes, FIB memory size of 100, 1000, 3000 entries (for f100, f1k, and f3k, respectively), and the Zipf α parameter values of 0.7 and 1.4 (for a0.7 and a1.4 respectively). Figures a and c consider the original CRoS-NDN scheme. Figures b and d consider the CRoS-NDN Tunnel Extension scheme. The simulation employs a rate of 50 interests per second per consumer.



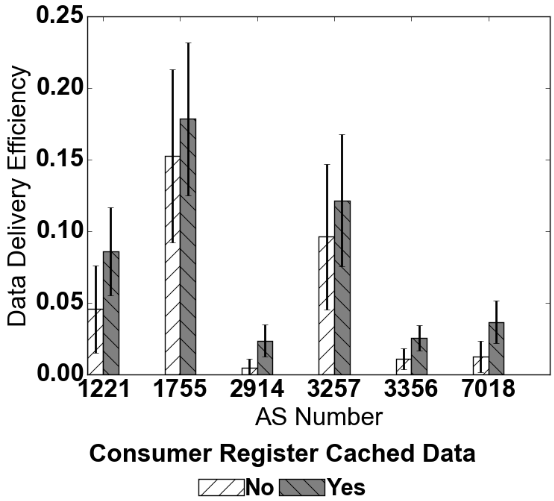
(a) Consumer rate variation for CRoS-NDN.



(b) Consumer rate variation for CRoS-NDN Tunnel Extension.



(c) Topology variation for CRoS-NDN.



(d) Topology variation for CRoS-NDN Tunnel Extension.

Figure A.4: Data delivery efficiency of CRoS-NDN and CRoS-NDN Tunnel Extension for consumer registration of data copies with cache sizes of routers with 100.000 entries.