

Enhancing Network Performance and Consumer Experience in Named Data Networking (NDN)

A Dissertation submitted to the University of Dublin, Trinity College

in fulfilment of the requirements for the degree of

Doctor of Philosophy (Computer Science)

June 2020

Declaration

I, the undersigned, declare that this work has not previously been submitted to this or any other University, and that unless otherwise state, it is entirely my own work.

Dated: June 29, 2020

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this Thesis upon request.

Dated: June 29, 2020

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor Dr Stefan Weber, for his continuous support towards the completion of my Ph.D and related research. His patience, motivation, and immense knowledge have guided me through the duration of my studies and helped me to maintain a high standard at each step of this thesis.

Besides my supervisor, I would also like to acknowledge the help of my colleagues, for their insightful comments and encouragement, and for the laborious questions which incentivised me to widen my research on various aspects.

My sincere acknowledgments go to my former employer and supervisor at Palo Alto Research Centre (PARC), Glenn Edens and Ignacio Solis, who provided me with the opportunity to join their group as a research intern, and helped me to understand and to further consider the practical aspects of my research.

An enormous thanks goes to all of my friends for supporting me spiritually throughout the accomplishment of this thesis. Our time together has certainly been valuable.

Last but not the least, my deepest gratitude goes to my beloved family, who has supported all my choices with endless love, understanding and patience.

I hope research stays true and motivated by personal curiosity to the world.

University of Dublin, Trinity College

June 2020

Publications Related to this Thesis

- Andriana Ioannou & Stefan Weber. *Towards on-path caching alternatives in Information-Centric Networks*. Published in Proceedings of the 39th IEEE Conference on Local Computer Networks (LCN), Edmonton, Canada, September 2014, pages 362-365.
- Andriana Ioannou & Stefan Weber. *Exploring content popularity in Information-Centric Networks*. Published in the China Communications Magazine, July 2015, Volume 12, Issue 7, pages 13-22.
- Andriana Ioannou & Stefan Weber. *A Survey of Caching Policies and Forwarding Mechanisms in Information-Centric Networking*. Published in the IEEE Communications Surveys and Tutorials, May 2016, Volume 18, Issue 4, pages 2847-2886.

Abstract

Named Data Networking (NDN) is an *Information-Centric Networking (ICN)* architecture proposed as an alternative to the current Internet infrastructure. Using the publish-subscribe paradigm and a standardised naming scheme, ICN allows the components of a publish-subscribe solution to interact without tight coupling between them.

NDN supports a hierarchical naming scheme that identifies chunks of an object resource such as a service, web page, file, etc., using a *content identifier*. Content identifiers should be location-free and all identical contents should share the same content identifier. This way, content can be freely distributed and cached within the infrastructure utilising the capacity of routers on the path from a source to a destination, called *on-path caching*. A hierarchical naming scheme may also enable the functionality of *request aggregation* at routers. The purpose of this mechanism is to aggregate close-in-time requests for the same content and to propagate only the first of them to a content source(s).

This thesis focuses on the exploration of the on-path caching feature and the request aggregation mechanism of NDN to enhance the network performance and the consumer experience within an Internet Service Provider (ISP)/Autonomous System (AS) network by reducing the intra and inter-network traffic and the content delivery times of consumers. By reducing the intra and inter-network traffic of an ISP/AS network, a

reduction of the content delivery times of consumers is also expected.

Popularity and Location-based Caching (PLbC) is a lightweight caching algorithm that utilises the criterion of content popularity and the criterion of the location of routers on delivery paths by incorporating them into the caching decision process to construct a probability. PLbC has been shown to outperform the caching policy proven to perform the best among a number of caching policies evaluated in this thesis, i.e. PC+. PLbC has been shown to yield a 3-6% higher probability of retrieving the content locally from the caches of routers compared to PC+, while caching 11-15% less content within the local ISP/AS network. Yet, PLbC is unable to fully utilise the cache capacity of routers as this approaches the catalog size, i.e. the number of object resources within a network.

Content Sharing-Extended Request Aggregation (CS-ERA) is a mechanism that focuses on content sharing between consumers that request to retrieve the same object resource. For this purpose, CS-ERA exploits the mechanism of request aggregation to bound successive requests to be satisfied within the boundaries of the local ISP/AS network, if a local replica exists. To increase the number of successive requests to be bounded, CS-ERA extends the naming granularity to which request aggregation is applied from a chunk to an object resource. CS-ERA has been shown to outperform the original request aggregation mechanism by aggregating 11-17% more object requests within an ISP/AS network, while satisfying 52-81% more chunks of an object resource locally. The results vary depending on whether on-path caching is enabled or disabled.

Contents

Acknowledgements	iv
Abstract	vi
List of Tables	xv
List of Figures	xvi
Glossary	xix
Chapter 1 Introduction	1
1.1 Information-Centric Networking (ICN)	3
1.2 Named Data Networking (NDN): An Overview	5
1.3 Problem Domain and Challenges	7
1.3.1 Challenges in On-path Caching	8
1.3.2 Challenges in Request Aggregation	9
1.4 Proposed Solutions	9
1.4.1 PLbC: Popularity and Location-based Caching	10
1.4.2 CS-ERA: Content Sharing-Extended Request Aggregation	11
1.5 Roadmap	12

1.6	Summary	12
Chapter 2	Background & State-of-the-Art	14
2.1	Named Data Networking (NDN): A Description	15
2.1.1	Naming	15
2.1.2	Interest and Data Packets	16
2.1.2.1	Type-Length-Value (TLV) Packet Format	16
2.1.2.2	Interest Packet	17
2.1.2.3	Data Packet	19
2.1.3	Content Store (CS)	24
2.1.4	Pending Interest Table (PIT)	25
2.1.5	Forwarding Information Base (FIB)	27
2.1.6	Upstream Process	27
2.1.7	Downstream Process	29
2.2	On-path Caching: Categorisation	30
2.2.1	Location-based Caching	33
2.2.2	Popularity-based Caching	36
2.2.2.1	Threshold-based Caching	37
2.2.2.2	Correlativity-based Caching	39
2.2.3	Probability-based Caching	40
2.2.3.1	Fixed-probability Caching	41
2.2.3.2	Dynamic-probability Caching	42
2.2.3.2.1	ProbCache (PC)	42
2.2.3.2.2	Hop-based Probabilistic Caching (HPC)	44
2.2.3.2.3	Congestion-Aware Caching (CAC)	46

CONTENTS

2.2.4	The Evolution	48
2.3	On-path Caching: Quantitative Comparison	52
2.3.1	ndnSIM Simulator	52
2.3.2	Caching Policies	53
2.3.2.1	ProbCache+ (PC+)	53
2.3.2.2	Hop-based Probabilistic Caching (HPC)	55
2.3.2.3	Congestion-Aware Caching (CAC)	55
2.3.2.4	Leave Copy Everywhere (LCE)	56
2.3.2.5	Leave Copy Last (LCL)	57
2.3.2.6	No Caching (NC)	57
2.3.3	Network Topologies	57
2.3.4	Applications	60
2.3.4.1	Consumer	60
2.3.4.2	Producer	64
2.3.5	Architectural Components	64
2.3.6	Evaluation Metrics	66
2.3.6.1	Cache-based Metrics	66
2.3.6.2	Network-based Metrics	67
2.3.7	Evaluation Results	68
2.3.8	Lessons Learned	80
2.3.9	Discussion	83
2.4	Request Aggregation: The Mechanism	86
2.4.1	Efficiency Analysis	87
2.4.2	Discussion	89
2.5	Summary	90

Chapter 3 PLbC: A Popularity and Location-based Caching Policy for	
On-path Caching	91
3.1 Context and Motivation	93
3.2 Design and Implementation	95
3.2.1 Statistical Information Table (SIT)	95
3.2.2 Utility Function and Caching Decision	98
3.2.3 Interest and Data Packets	102
3.2.4 PIT Entry Format	105
3.2.5 Upstream Process	106
3.2.6 Downstream Process	108
3.3 System Model	109
3.3.1 Applications	110
3.3.2 Architectural Components	110
3.4 Evaluation	111
3.4.1 Evaluation Results	112
3.5 Discussion	120
3.5.1 Lessons Learned	121
3.5.2 Evaluation Limitations	124
3.5.3 Design Limitations	126
3.5.4 Security Implications	127
3.6 Summary	130
Chapter 4 CS-ERA: A Mechanism for Content Sharing and Inter-network	
Traffic Reduction	131
4.1 Context and Motivation	133

CONTENTS

4.2	Design and Implementation	135
4.2.1	CS-ERA Border Router	135
4.2.2	Interest and Data Packets	137
4.2.3	PIT Entry Format	141
4.2.4	Upstream Process	143
4.2.5	Downstream Process	145
4.2.6	Principal Consumer	146
4.2.7	FIB and Forwarding Strategy	149
4.3	System Model	150
4.3.1	Applications	150
4.3.2	Architectural Components	152
4.4	Evaluation	154
4.4.1	Evaluation Metrics	156
4.4.2	Evaluation Results	156
4.5	Discussion	170
4.5.1	Lessons Learned	170
4.5.2	Evaluation Limitations	172
4.5.3	Design Limitations	174
4.5.4	Deployment Implications	176
4.5.5	Security Implications	177
4.5.6	Request Aggregation vs. On-path Caching	179
4.6	Summary	182
Chapter 5 Conclusions and Future Work		184
5.1	Objectives and Summary	184

5.1.1	PLbC	185
5.1.2	CS-ERA	186
5.2	Contributions	188
5.3	Future Work	189
5.3.1	PLbC	190
5.3.2	CS-ERA	191
Bibliography		194

List of Tables

1.1	A comparison of the CDN, P2P and ICN content-centric technologies. . .	3
2.1	Summary of the caching policies reviewed in this thesis, categorised with regard to the caching level and type of cooperation criteria.	48
2.2	Summary of the system model and the evaluation parameters used in the evaluation of the PC+, HPC, CAC, LCE and LCL caching policies. . . .	61
4.1	Summary of the system model and the evaluation parameters used in the evaluation of the PC+, HPC, CAC, LCE and LCL caching policies. . . .	153

List of Figures

1.1	Representation of the publish-subscribe paradigm.	4
1.2	Representation of the NDN architecture.	6
2.1	Representation of the Interest packet format in NDN.	18
2.2	Representation of the Data packet format in NDN.	22
2.3	Upstream forwarding process of an Interest at an NDN router.	28
2.4	Downstream forwarding process of Data at an NDN router.	29
2.5	Representation of the PCP caching policy.	34
2.6	Representation of the HVC caching policy.	35
2.7	Representation of the MPC caching policy.	38
2.8	Representation of the PbLRU caching policy.	39
2.9	Calculation of the x and y values used in dynamic-probability calculations.	41
2.10	Representation of the PC caching policy.	43
2.11	Representation of the HPC caching policy.	45
2.12	Representation of the CAC caching policy.	47
2.13	Taxonomy of the caching policies reviewed in this thesis according to the caching criteria used.	51
2.14	Representation of the Tiscali Topology (AS-3257) used in the simulations.	58

LIST OF FIGURES

2.15	Representation of the Exodus Topology (AS-3967) used in the simulations.	59
2.16	Representation of the mean hop-count and SD values of an approach for the AS-3257 network topology.	69
2.17	Representation of the mean hop-count and SD values of an approach for the AS-3967 network topology.	70
2.18	Representation of the mean CHR and SD values of an approach for the AS-3257 network topology.	73
2.19	Representation of the mean CHR and SD values of an approach for the AS-3967 network topology.	74
2.20	Representation of the mean CF and SD values of an approach for the AS-3257 network topology.	76
2.21	Representation of the mean CF and SD values of an approach for the AS-3967 network topology.	77
2.22	Representation of the mean CER and SD values of an approach for the AS-3257 network topology.	78
2.23	Representation of the mean CER and SD values of an approach for the AS-3967 network topology.	79
3.1	Abstract representation of the functionality of the PLbC caching policy. .	96
3.2	Representation of the Interest packet format for the operation of PLbC. .	102
3.3	Representation of the Data packet format for the operation of PLbC. . . .	104
3.4	Upstream forwarding process of an Interest at an NDN router that utilises the PLbC caching policy.	107
3.5	Downstream forwarding process of Data at an NDN router that utilises the PLbC caching policy.	108

3.6	Representation of the mean hop-count and SD values of an approach for the AS-3257 network topology.	113
3.7	Representation of the mean hop-count and SD values of an approach for the AS-3967 network topology.	114
3.8	Representation of the mean CHR and SD values of an approach for the AS-3257 network topology.	115
3.9	Representation of the mean CHR and SD values of an approach for the AS-3967 network topology.	117
3.10	Representation of the mean CF and SD values of an approach for the AS-3257 network topology.	118
3.11	Representation of the mean CF and SD values of an approach for the AS-3967 network topology.	119
3.12	Representation of the mean CER and SD values of an approach for the AS-3257 network topology.	121
3.13	Representation of the mean CER and SD values of an approach for the AS-3967 network topology.	122
4.1	Abstract representation of the functionality of the CS-ERA mechanism.	133
4.2	Representation of the Interest NACK packet format in NDN.	138
4.3	Representation of the Data packet format for the operation of CS-ERA.	140
4.4	Upstream forwarding process of an Interest at a CS-ERA border router.	144
4.5	Downstream forwarding process of Data at a CS-ERA border router.	146
4.6	Representation of the mean AF and SD values of an approach for the AS-3257 network topology, when caching is disabled.	157

LIST OF FIGURES

4.7	Representation of the mean AF and SD values of an approach for the AS-3967 network topology, when caching is disabled.	158
4.8	Representation of the mean object requests issued by a consumer and SD values for the AS-3257 network topology, when caching is disabled.	160
4.9	Representation of the mean object requests issued by a consumer and SD values for the AS-3967 network topology, when caching is disabled.	161
4.10	Representation of the mean AER and SD values of an approach for the AS-3257 network topology, when caching is disabled.	162
4.11	Representation of the mean AER and SD values of an approach for the AS-3967 network topology, when caching is disabled.	163
4.12	Abstract representation of the functionality of ORIGINAL when single request aggregations occur.	164
4.13	Representation of the mean AF and SD values of an approach for the AS-3257 network topology, when caching is either disabled or enabled.	166
4.14	Representation of the mean AF and SD values of an approach for the AS-3967 network topology, when caching is either disabled or enabled.	167
4.15	Representation of the mean AER and SD values of an approach for the AS-3257 network topology, when caching is either disabled or enabled.	168
4.16	Representation of the mean AER and SD values of an approach for the AS-3967 network topology, when caching is either disabled or enabled.	169

Glossary

Autonomous System (AS): Single network or group of networks under the same administration that defines a common routing policy to the Internet infrastructure.

Best Route (BR): Forwarding strategy in NDN that utilises the SPR protocol.

Border Router: Router within an ISP/AS network that propagates traffic to the Internet infrastructure. Border routers are not connected to consumers.

Cache Pollution: The storage of content that is requested once, thus does not exploit the existing cache capacity efficiently, while replacing more useful content.

Cache Pollution Attack: Type of DoS attack where an adversary aims to disrupt cache locality to increase link utilisation and cache misses at an NDN router.

Caching Policy: Policy used by an NDN router to cache new content.

Catalog Size: The number of individual object resources within an ISP/AS network.

Chunk: Part/piece of an object resource defined by an application.

Consumer: Entity in NDN that requests to retrieve content.

Content-Centric Networking (CCN): Commercial ICN architecture.

Content Delivery Network (CDN): Clusters of servers distributed throughout the Internet infrastructure, used to replicate content from content providers.

Content Delivery Time: The time interval between the expression of a request for

an object resource and until the retrieval of this object resource is complete.

Content Identifier: Name used to identify a chunk of an object resource.

Content Notification Service: Service used in a publish-subscribe paradigm to map a content identifier/name/prefix to a content source.

Content Request: Request for content on a chunk naming granularity.

Content Reply: Delivery of content on a chunk naming granularity.

Content Service Provider (CSP): Organisation that owns or it is licensed to sell and distribute content in the Internet infrastructure.

Content Sharing-Extended Request Aggregation (CS-ERA): Request aggregation mechanism applied to object resources to enable content sharing between consumers.

Content Store (CS): Data structure in NDN that stores content.

CS-ERA Border Router: Border router that supports the functionality of CS-ERA.

Data: Type of packet in NDN used to identify a chunk of an object resource.

Denial-of-Service (DoS): Maliciously intended disruption of a service or the availability of a network resource to its intended, legitimate users.

Domain Name Server (DNS): Structure that maps a domain to an IP address.

Downstream Process: The arrival of Data at an NDN router.

Edge Router: Router connected to consumers and the last router on a delivery path.

End-to-end Communication Model: The communication model used in the Internet infrastructure that transfers data between one or more devices within a network.

Exponential Moving Average (EMA): Type of a moving average, where more weight is given to the latest observation.

Face: Extended version of an interface, that may be used to represent the communication between the network infrastructure and an application process at an NDN router.

Forwarding Information Base (FIB): Data structure in NDN that keeps track of

the faces advertised for a content identifier/name/prefix within an ISP/AS network.

FIB Pollution Attack: Type of DoS attack where an adversary aims to disrupt the functionality of a FIB to install optimal routes to increase link utilisation.

Information-Centric Networking (ICN): An alternative Internet architecture where content is addressable and routable through the use of a naming scheme.

Interest: Type of packet in NDN used to identify a request for the retrieval of a chunk.

Interest Flooding Attack (IFA): Type of DoS attack where an adversary generates a large number of close-in-time successive Interests to overflow the PIT of an NDN router.

Interest NACK: Type of Interest that NDN routers may propagate downstream if they are unable to propagate an Interest upstream to a content source.

Intermediate Router: A router other than an edge router and/or a border router.

Internet: A worldwide network of computer networks.

Internet Protocol (IP): Numerical identifier that uniquely identifies a device within a network, assigned to enable the communication with other devices.

Internet Service Provider (ISP): Organisation that provides services to individuals and other organisations/companies to access the Internet.

Locality-First Best Route (LFBR): Forwarding strategy in NDN able to distinguish among faces that refer to local replicas of content and permanent content sources.

Multi-homing: A practice used to connect a device or a computer network to more than one networks, to increase reliability and enhance performance.

Multi-path Forwarding: Forwarding technique used to propagate traffic through a network using more than one forwarding paths.

Named-data Link State Routing (NLSR): Name-based routing protocol in NDN that disseminates reachability to name prefixes.

Named Data Networking (NDN): ICN architecture that uses a hierarchical naming

scheme. NDN is the academic equivalent of CCN.

Name Resolution Service (NRS): Structure that maps names to a content source.

Naming Granularity: The type of content to which a name refers.

Naming Scheme: Scheme that uniquely identifies content through the Internet infrastructure utilising name content identifiers/names/prefixes.

ndnSIM: ns-3-based, open-source simulator for the NDN architecture.

Network Manager Module (NMM): Entity that has knowledge of the ISP/AS network topology and traffic.

Object Request: Request for the retrieval of an object resource, indicated by the reception of a content request for the first chunk of an object resource.

Object Resource: Type of content such as services, web pages, files, videos, songs etc.

Off-path Caching: Feature of ICN that caches content at the edge nodes of a network.

On-path Caching: Feature of ICN that caches content at the routers on delivery paths.

One-timer Objects: Object resources that are requested once.

Outstanding Interests: Interests that have been forwarded to a content source(s), but have not been satisfied yet.

Peer-to-Peer Network (P2P): Group of end users that share content using a content directory service that maps content requests to potential sources.

Pending Interest Table (PIT): Data structure in NDN that keeps track of the outstanding Interests.

Popularity and Location-based Caching (PLbC): Caching policy based on the criteria of content popularity and the location of routers on delivery paths.

Prefix: Name identifier that identifies an object resource.

Prefix-Link State Advertisement (Prefix-LSA): Routing messages that carry routing information about name prefixes in NSLR.

Principal Consumer: Consumer responsible for the retrieval of an object resource.

Producer: Entity in NDN that produces content.

Publisher: Entity in a publish-subscribe system that announces content to a content notification service.

Quality-of-Service (QoS): A network's ability to provide better services to organisations/companies and/or entities.

Request Aggregation Mechanism: Mechanism in NDN that aggregates close-in-time Interests for the same content and propagates only one of them to a source.

Request Band: Mechanism similar to a request aggregation mechanism, where successive Interests are merged into a single Interest.

Sequence Number: The number of a chunk of an object resource.

Shortest Path Routing (SPR): Routing protocol that calculates the paths with the shortest distance/cost between two devices/entities.

Single-Interest Object Resource: Object resource with one chunk to be retrieved.

Stateful Forwarding: Type of forwarding based on the functionality of a PIT.

Stateless Forwarding: Type of forwarding unrelated to the functionality of a PIT.

Statistical Information Table (SIT): Data structure in NDN that collects information about the popularity of object resources, indicated by the number of object requests.

Subordinate Consumer: Consumer whose Interest was aggregated by CS-ERA.

Subscriber: Entity in a publish-subscribe system that contacts a content notification service to retrieve content.

Successive Interest: Interest that refers either to the same chunk of an object resource as a previous-in-time Interest or the next sequential chunk of an object resource.

Symmetric Forwarding: Type of forwarding, where both content requests and content replies follow the same set of routers.

Temporary Request Aggregation: The occurrence of request aggregations of Interests that may happen once or occasionally during the retrieval of an object resource.

Trust Model: Set of rules that define a relationship between content identifiers/names and public-private key pairs used by producers and consumers to sign and verify content.

Type-Length-Value (TLV): Type of packet format used in NDN to represent packets.

Upstream Process: The arrival of an Interest at an NDN router.

User Generated Content (UGC): Type of content created by consumers/individuals that is publicly available in the Internet infrastructure.

Chapter 1

Introduction

The original design of protocols for the Internet focused on the end-to-end communication model for the exchange of information between a client and a server [Saltzer]. However, usage patterns and technologies have changed since these protocols were developed and today's Internet usage is dominated by the distribution and retrieval of content instead of the connection to specific servers [Feldmann]. This mismatch of traditional protocols and current usage patterns results in a number of difficulties with regard to content availability, multi-homing and mobility for both clients and servers, while raising scalability and performance concerns for various mechanisms [Balakrishnan , Jacobson]. The gap between the original protocols and their current use is bound to increase as a number of reports on the usage patterns of the Internet have associated an increased popularity of multimedia applications with the rise of the Internet traffic, i.e. 3.3 zettabytes by 2021 [Gantz]. Almost 80% of this traffic is expected to be video [Sandvine , Cisco].

Content Delivery Networks (CDNs) [Buyya] and Peer-to-Peer (P2P) [Androutsellis-Theotokis] networks aim to shift the focus of the end-to-end communication model on content, by providing retrieval mechanisms as overlays and by exploiting cheap storage

and processing capabilities within the existing Internet infrastructure.

- **CDNs:** CDNs have been conceived to serve application-layer content, such as web pages, to clients from servers that are closest to the client's physical location. Their operation is based on clusters of servers distributed throughout the Internet infrastructure and the replication of content from content providers to these clusters. The selection of servers that will host the replicas is based on monitoring and load-balancing operations, while considering the Quality-of-Service (QoS) guarantees that content providers request [Passarella]. Based on the physical location of clients, requests are directed through the Domain Name Servers (DNS) of the CDN provider towards the closest servers that hold replicas of the requested content.
- **P2Ps:** P2P networks are based on the exchange of content between end users; each end user acting as client, as well as server. P2P networks are more open concerning the traffic they serve, including files, movies, songs, etc., but they are limited in the sense that specific protocols are used for their operation [Lua , Passarella]. In P2P networks, a directory server resolves content requests into a number of potential sources called *peers*. In contrast to the replicas made in CDNs, peers in P2P networks may not have the full object requested but only parts of it, i.e. *chunks*. These chunks of a content resource may be defined by an application e.g. chunks 1 to 7 of a movie or by the P2P protocol. Thus, content requests may be propagated to one or more peers simultaneously, and depending on the number of peers reduce the overall retrieval time of content.

Despite the advantages of CDNs and P2P networks over the traditional network protocols, their performance and acceptance is limited due to their operation at the application layer, and the technological and commercial administration boundaries that

Characteristics	CDN	P2P	ICN
Infrastructure Type	Distributed/Clustered	Distributed	Distributed
Protocol Type	Proprietary	Application-specific	Standardised
Layer in OSI Stack	Application	Application	Network
Naming Type	Source-related	Source-free	Source-free
Naming Granularity	Object	Chunk	Object/Chunk
Infrastructure Scale	Proprietary Infrastructure	End-users	Internet Infrastructure

Table 1.1: A comparison of content-centric technologies shows the aspects derived by ICN from CDNs and P2P networks and the differences between these technologies.

they apply to [Passarella , Vakali]. As the original design of the Internet was conceived to support a basic set of services, future changes on its infrastructure should guarantee backwards compatibility. Therefore, both CDNs and P2Ps have been developed as overlays over the current Internet infrastructure, resulting in isolated deployments that do not share a common naming scheme or encoding of their data units.

1.1 Information-Centric Networking (ICN)

Information-Centric Networking (ICN) provides an alternative to the traditional end-to-end communication model of the current Internet architecture by focusing on information dissemination and information retrieval. ICN aims to address the shortcomings of CDNs and P2P networks by defining a common protocol - similar to the Internet Protocol (IP) - that can be used by various application-layer solutions, while exploiting the processing power and storage within the infrastructure to replicate content. Table 1.1 provides an overview of the characteristics of CDNs and P2P networks in comparison to the characteristics of ICN, demonstrating how some of these have been derived from CDNs and P2Ps, while others aim to address their shortcomings.

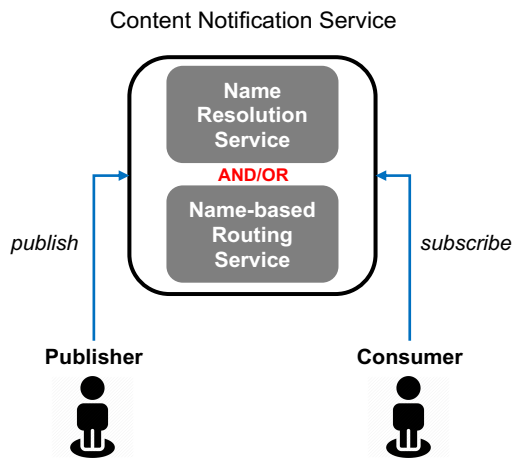


Fig. 1.1: The publish-subscribe paradigm, where publishers publish content to a content notification service and consumers subscribe to it to retrieve content.

ICN is based on the publish-subscribe paradigm and a standardised naming scheme that allows the components of a publish-subscribe solution to interact without the necessity of tight coupling between them [Ahlgren b, Xylomenos]. Fig. 1.1 illustrates how in a publish-subscribe paradigm *publishers* announce their content to a *content notification service* to make it available for subscription and *subscribers* contact the content notification service to retrieve content [Choi , Tyson]. A content notification service may consist of a *name resolution service* and/or a *name-based routing service* [Ahlgren b, Xylomenos]. ICN ensures space and synchronisation decoupling between publishers and subscribers by moving away from a traditional connection-based approach and relying on the underlying infrastructure to mediate between publishers and subscribers [Jacobson].

A naming scheme identifies *object resources* such as services, web pages, files and videos, or parts of an object resource defined by an application, i.e. chunks, using a *content identifier* or *name*. For the rest of this thesis, the terms content identifier and name will be used interchangeably. The type of content to which a content identifier

refers defines the *naming granularity* of the architecture, i.e. *object* or *chunk naming granularity*. In the design of naming schemes, a number of decisions need to be made: whether content identifiers should reveal structural information, whether content identifiers should be human-readable, whether content identifiers should be flat or hierarchical or self-certified [Ahlgren b, Xylomenos]. A naming scheme may affect both the scalability and the security aspects of the mechanisms supported by an ICN architecture and enable functionalities such as authentication, replication and flow-control [Adhatarao , Bari , Ghodsi a]. Content identifiers in ICN are considered to be unique and application-agnostic. Content identifiers should not involve information that would bind the content to a location and all identical contents should share the same content identifier [Balakrishnan , Mazieres , Walfisha]. If these constraints are met, the content can be freely distributed and cached within a network, enabling functionalities such as in-network caching, *request aggregation*, multi-homing and multi-path forwarding.

1.2 Named Data Networking (NDN): An Overview

The *Named Data Networking (NDN)* framework is an NSF-funded research project that implements various components of an ICN architecture [Zhang a]. In contrast to other ICN architectures such as DONA [Koponen], COMET [Garcia], PSIRP-PURSUIT [Fotiou] and NetInf-SAIL [Dannewitz], whose research activity has ended, the research activity of NDN is ongoing. NDN has also been adopted by numerous researchers for the evaluation of mechanisms proposed in the context of ICN, such as on-path caching and multi-path forwarding [Muscariello a, Tortelli , Mastorakis b]. To enable comparison between the research conducted in this thesis and research conducted by the NDN and ICN research communities, NDN is the ICN architecture adopted in this thesis.

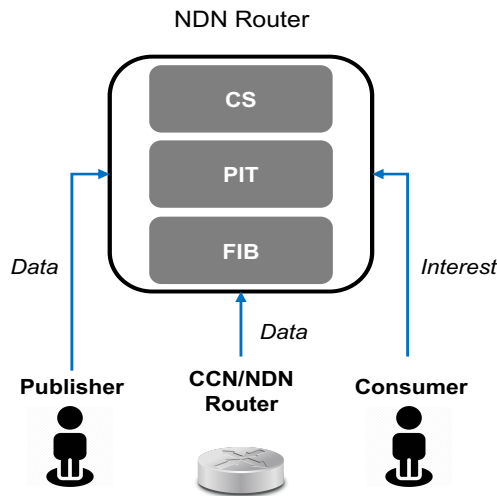


Fig. 1.2: An overview of the NDN architecture, consisting of the CS, PIT and FIB architectural components.

Fig. 1.2 presents the NDN architecture, which is based on three fundamental architectural components: a *Content Store (CS)*, a *Pending Interest Table (PIT)* and a *Forwarding Information Base (FIB)*. In NDN, the term *Interest* is used to refer to a content request. The term *Data* is used to refer to the delivery of content. Both Interests and Data contain content identifiers on a chunk naming granularity that specify the content they serve. Interests are issued by consumers/subscribers and they are propagated to a content source(s) based on the next-hop information maintained in the FIB of NDN routers. Data are issued by a content source/publisher or an NDN router that possesses a replica of the content in its CS. Data are subject to caching at each router on the delivery path(s), from a content source to a consumer. This introduces the problem of *on-path caching*, according to which an NDN router decides whether to cache Data based on a caching policy. An overview of the challenges that on-path caching raises is presented in Section 1.3.2. To deliver Data back to consumers, the PIT architectural

component of NDN routers is used to maintain information about the Interests that have been propagated to a content source(s). Consequently, both Interests and Data are forwarded through the same NDN routers and paths, a process called *symmetric forwarding* [Yi a,Zhang a]. Routers in NDN are also enabled with the mechanism of request aggregation, according to which Interests that refer to the same content are aggregated at the PIT of a router and only the first of them is propagated to a content source(s).

A thorough description of the NDN architecture is given in Chapter 2, including the Interest and Data packet formats, and the CS, PIT and FIB architectural components. A description of the *upstream process* and the *downstream process* followed at an NDN router is also given. The term upstream process refers to the arrival of Interests at an NDN router and their propagation to a content source(s). The term downstream process refers to the arrival of Data and their propagation to a consumer(s).

1.3 Problem Domain and Challenges

The problem domain of this thesis lies on the exploration of the on-path caching feature and the request aggregation mechanism of NDN to enhance the network performance and the consumer experience within an NDN network through the following objectives:

- i By reducing the traffic within an Internet Service Provider (ISP) or an Autonomous System (AS) network, and the traffic propagated beyond the boundaries of this network to the Internet infrastructure.
- ii By reducing the content delivery times experienced by consumers for the retrieval of an object resource. Taking into account that the content delivery times of consumers may be determined by the number of hops traversed by both Interests

and Data, a reduction of the intra and inter-network traffic of an ISP/AS network is also expected to result to a reduction of the content delivery times of consumers. Consequently, this objective of the thesis is complementary to the first objective.

1.3.1 Challenges in On-path Caching

Due to the integration of on-path caching with the forwarding mechanism, a caching decision is limited to the content that has been requested by an Interest and the number of routers on the delivery path(s). Alternative approaches to these limitations necessitate the creation of replicas and mechanisms to propagate these replicas to the required locations at a given time. Examples of such on-path caching policies have been shown to introduce more traffic and complexity within a network [Li e, Ming , Wang b].

An additional challenge that derives from the integration of on-path caching with the forwarding mechanism is the time required to perform a number of operations, such as a caching decision, a replacement decision and a lookup in the cache of a router [Ahlgren a, Rossi a]. The rationale behind this claim is that the time required for each operation adds to the content delivery times experienced by consumers. Given that one of the objectives of this thesis is the reduction of this metric, all operations need to be bound by an upper time-limit, while the occurrence of timeouts and Interest retransmissions should be avoided. Thus, lightweight caching policies and replacement policies are preferable to the use of time intensive and complicated alternatives. To bound the lookup time in the cache of a router, a cache size between 10GB and 32GB has been suggested [Arianfar , Lee].

Another challenge that derives from the integration of on-path caching with the forwarding mechanism is the management of large *catalog sizes* [Ioannou ,Katsaros ,Rossi a], where the term catalog size refers to the number of object resources available within a network. The catalog size for ICN is estimated to 10^{12} object resources [Ghodsi b,

Detti a, D’Ambrosio]. This challenge is expected to worsen since on-path caching does not follow any particular structure and can be applied to any router within a network equipped with a cache, e.g. *scale-free topologies* [Barabasi , Bollobas] and *Internet-like topologies* derived from the Internet infrastructure [Caida , Spring]. This structure-free nature of on-path caching makes structurally dependent caching policies impractical.

1.3.2 Challenges in Request Aggregation

A request aggregation mechanism has been defined to aggregate Interests that refer to the same chunk of an object resource and prevent their propagation upstream. Consequently, as request aggregation mechanism constitutes a native flow-control mechanism and a security mechanism of the NDN architecture that aims to reduce the amount of traffic within a network and prevent the occurrence of Denial-of-Service (DoS) attacks [Ghali]. Nonetheless, its functionality is bound to the level of chunk naming granularity and the time-interval for which information remains alive in a PIT. Given the fact that this time is expected to be of the order of tens to hundreds of milliseconds, the probability of request aggregation occurring within a network, thus the efficiency of this mechanism, is expected to be limited as well [Dabirmoghaddam , Ghali , Carofiglio a].

1.4 Proposed Solutions

The objectives stated in this thesis refer to the enhancement of the network performance and the consumer experience within an NDN network. To this end, the exploration of the on-path caching feature and the request aggregation mechanism are investigated. A description of the solutions proposed is presented in the following sections.

1.4.1 PLbC: Popularity and Location-based Caching

The first solution proposed in this thesis relates to the placement of replicas on delivery paths during the downstream process, i.e. the on-path caching problem. The goal is to investigate the benefits of a dynamic-probability caching policy, called *Popularity and Location-based Caching (PLbC)* policy, composed of the *Popularity* factor (P_f) and the *Location* factor (L_f), formulated as follows: $p = PLbC = P_f \times L_f$.

- **P_f** : The P_f factor refers to the popularity of an object resource o at a router to which a content refers, calculated by the number of the object requests received ctr_o , and compared against the maximum number of object requests for any object $max_{\forall o \in O} ctr_o$ to construct a probability, i.e. $ctr_o / max_{\forall o \in O} ctr_o$. The definition of the P_f factor has been based on the experimental evaluation of a former definition given by the researcher. According to this former definition, the number of the object requests received ctr_o was compared against the total number of object requests to construct a probability, i.e. $ctr_o / \sum \forall o \in O ctr_o$. This definition has been shown to result in a significant underutilisation of the cache capacity of routers since the values of the P_f factor were considerably low, e.g. $< 10^{-4}$. Hence, an alternative approach has been adopted to enforce the comparison between contents. This approach is expected to increase the overall caching probability.
- **L_f** : The L_f factor refers to the reduction of the number of hops to be traversed by successive Interests and Data by caching a replica of the received content at the local NDN router. The definition of the L_f factor has been based on the hp_u and the hp_d metrics included in the Interest and Data packets formats to construct a probability, i.e. $L_f = hp_d / hp_u$. The hp_u metric refers to the number of hops traversed by an Interest from the consumer that issued the Interest to an NDN

router that possesses a replica of the content in its CS or a content source. The hp_d metric refers to the number of hops traversed by the corresponding Data to reach the router that performs the caching decision since the packet was issued.

By combining the P_f and L_f factors, PLbC aims to answer the questions of "what content to cache" and "where to cache it". By combining the P_f and L_f factors to construct a probability, PLbC aims to be flexible and reactive to the dynamic environment of ICN, yet simple to stay within the time boundaries defined in Section 1.3.1.

1.4.2 CS-ERA: Content Sharing-Extended Request Aggregation

The second solution proposed in this thesis relates to the functionality of a request aggregation mechanism. The goal is to investigate the benefits of extending the functionality of the original request aggregation mechanism into a content sharing mechanism, called *Content Sharing-Extended Request Aggregation (CS-ERA)*. CS-ERA focuses on content sharing between consumers that request to retrieve the same object resource. To do so, CS-ERA exploits the mechanism of request aggregation to bound successive Interests to be satisfied locally within an ISP/AS network, if a local replica exists. To enable communication between consumers and the satisfaction of successive Interests within the local ISP/AS network, both the Interest and Data packet formats have been modified to include additional information. To increase the number of Interests to be bounded, CS-ERA extends the naming granularity to which request aggregation is applied from a chunk to an object resource. This way, CS-ERA increases the probability of request aggregation occurring within a network since any Interest that refers to either chunk of an object resource may be aggregated instead of a single chunk of an object resource.

1.5 Roadmap

The remainder of this thesis is structured as follows: In chapter 2, a detailed description of the NDN architecture is given, followed by a State-of-the-Art (SOA) description of the on-path caching feature and the request aggregation mechanism reviewed in this thesis. A quantitative comparison between a number of on-path caching policies based on simulations is also provided. The conclusions derived from this chapter will influence the design of the PLbC caching policy and the CS-ERA mechanism presented in the following chapters towards satisfying the objectives of this thesis. In chapter 3, the PLbC caching policy is introduced. In chapter 4, the CS-ERA mechanism is introduced. In chapter 5, the thesis concludes by stating the contribution and the fulfilment of the objectives defined, while suggesting directions for future work.

1.6 Summary

In this chapter, Information-Centric Networking (ICN), an alternative to the traditional end-to-end communication model of the current Internet infrastructure has been presented. Using the publish-subscribe paradigm and a naming scheme to “address” content, ICN architectures aim to define a common protocol - similar to IP - that can be used by various application-layer solutions to disseminate and retrieve content, while exploiting the processing power and storage within the infrastructure to replicate content.

One of the most significant ICN architectures, NDN, has been presented. The term NDN refers to the Named Data Networking (NDN) ICN architecture, the ICN architecture adopted in this thesis. To this end, a brief description of the NDN architecture has been provided to ease the description on the functionality and the challenges of the feature of on-path caching and the mechanism of request aggregation, i.e. the main

areas of focus of this thesis. Last, the solutions proposed in this thesis to enhance the network performance and the consumer experience within an NDN network, i.e. the Popularity and Location-based Probabilistic Caching (PLbC) policy and the Content Sharing-Extended Request Aggregation (CS-ERA) mechanism have been introduced.

Chapter 2

Background & State-of-the-Art

The context of this chapter is threefold and may be described as follows:

- A detailed description of the NDN architecture is given, including the naming scheme adopted, and the Interest and Data packet formats. Moreover, a description of the CS, PIT and FIB architectural components is given, followed by a description of the upstream process and the downstream process at an NDN router.
- A detailed description of the on-path caching policies reviewed in this thesis is given, highlighting their advantages and disadvantages. Moreover, a taxonomy that highlights the evolution of the caching criteria used and the caching policies is constructed. To provide an intuition of the performance of a caching policy, and the effect of the caching criteria on the performance of a network and the experience of consumers, a number of caching policies reviewed are also evaluated based on simulations. The conclusions derived from this evaluation will help the researcher to construct the dynamic-probability caching policy proposed in chapter 3.
- A detailed description of the operation of a request aggregation mechanism is given,

followed by a review on the efficiency of a request aggregation mechanism and a discussion on the complications encountered from its operation. The conclusions derived from this discussion will help the researcher to construct the mechanism proposed in chapter 4.

2.1 Named Data Networking (NDN): A Description

ICN is a content-based communication model, whereby network traffic is identified by the content itself using content identifiers, instead of the location of content. Due to this content and location decoupling, ICN provides a clean slate support to content-based functionalities such as on-path caching and request aggregation, the two areas of focus of this thesis. While on-path caching is independent of the ICN architecture, request aggregation applies strictly to *stateful forwarding* ICN architectures that keep track of the content requests propagated upstream within a network, i.e. the NDN and CCN architectures. NDN is the academic equivalent of CCN and the ICN architecture adopted in this thesis, a detailed description of which is given in this section.

2.1.1 Naming

Besides the prerequisites of naming schemes defined in Section 1.1, names in NDN are hierarchically constructed. A common approach used to ease the construction of names is by adopting human-readable URI-like strings, where the delimiter “/” is used to distinguish between name components [Team 14]. Hierarchical names are useful for applications to represent relationships between object resources and between pieces of an object resource, i.e. versions and chunks, and to ease the operation of routing protocols. As an example, the following hierarchical name *ndn:/object1/v1/s0* may be used to refer

to the 1st chunk ($s0$) of the 1st version ($v1$) of the object resource $ndn:/object1$. The aforementioned name has been defined on a chunk naming granularity. Nonetheless, content identifiers that refer to an object naming granularity, also called *prefix*, are constructed no different to the above. The only difference is that a prefix should not include any sequence numbers (s). If such a number exists, it should be discarded. Following the definition of the name $ndn:/object1/v1/s0$, the equivalent prefix is: $ndn:/object1/v1$.

To convert human-readable URI-like strings into the equivalent byte representation, the UTF-8 encoding scheme is used. The UTF-8 encoding scheme restricts certain bytes to be presented in a UTF-8 encoded string. Such bytes are used to define the meaning of specific characters/strings within the original NDN URI-like string, including the version of a content or an object resource v , and the chunk or *sequence number* (s or *seq*).

2.1.2 Interest and Data Packets

In this section, a description of the Interest and Data packet formats is given, as these have been defined in the documentation of the NDN architecture [Project a]. At this point it is worth noting, that only the fields of a packet necessary and/or related to the context of this thesis are presented. This is in order to avoid confusing the reader with unnecessary information, since a number of fields in either packet format is optional.

2.1.2.1 Type-Length-Value (TLV) Packet Format

NDN packets are formatted using the Type-Length-Value (TLV) packet format, where a packet is defined as a collection of nested TLVs maintained in the first and outermost TLV, called TLV_0 . Consequently, NDN packets have neither a fixed packet header format, nor a protocol version number, a design convention that eases the modification of packets over time. A TLV is defined by the *Type*, *Length* and *Value* fields, where:

- **Type** is a number within the range $\{1, 2^{32} - 1\}$, used to indicate the type of information maintained within a TLV. Consequently, the size of this field is defined to be equal to 1-4 bytes. The value of Type should be unique within each nested TLV. A value of Type equal to 0 has been reserved to indicate an invalid TLV.
- **Length** is the number of bytes used by Value. The size of this field is defined to be equal to 1-8 bytes.
- **Value** is a variable-sized collection of bytes used to represent the information maintained within a TLV. Hence, the size of this field can not be defined a priori.

Both Type and Length fields are encoded using the same variable-size encoding scheme to ease deployment, where the first byte is used to indicate the byte of the encoding scheme in which the value of each field is included.

2.1.2.2 Interest Packet

For the purpose of this thesis, an Interest packet is described as follows:

- **name:** A *name* is the only mandatory field in an Interest packet that represents the name content identifier of an Interest. A name in NDN is a hierarchical identifier composed of a number of name components/TLVs, including the *GenericNameComponent* and additional optional name components that may be used to restrict the Data to be retrieved. An example of such restriction would be to limit an Interest issued to retrieve Data published by a particular publisher. In contrast to its additional name components, a *GenericNameComponent* has no restriction on the type of its value field. Consequently, the size of a name can not be bounded and it is dependent to the size of the individual TLVs contained. Interests are

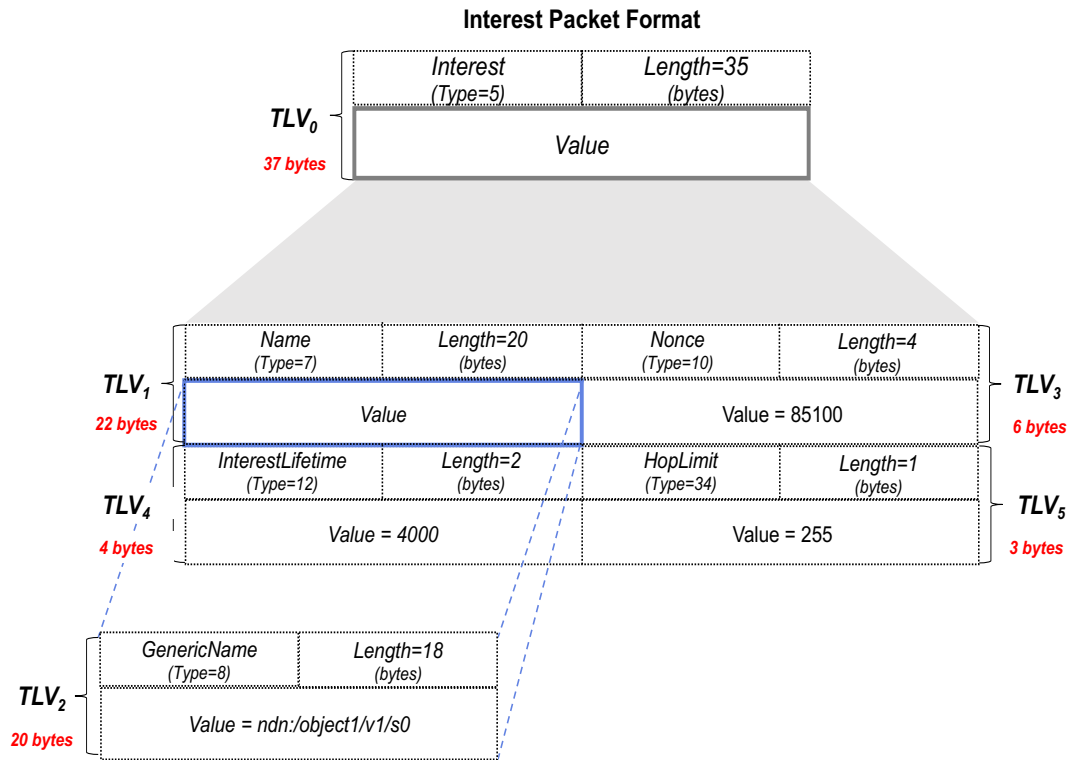


Fig. 2.1: Representation of the fields of an Interest packet necessary and/or related to the context of this thesis for the retrieval of the `ndn:/object1/v1/s0` chunk/content.

restricted to contain at least one name component to identify the content to be retrieved. Interests with an empty name field will be discarded by the forwarder.

- **nonce:** A *nonce* is an optional field in an Interest packet, yet a mandatory field when the Interest is propagated within a network. A nonce is a random numeric identifier generated by an application, that combined with the name content identifier of an Interest uniquely identifies an Interest within a network. A nonce is used to detect duplicated Interests that have been forwarded through multiple faces within a network and discard them. Nonces are represented using 4-byte long byte-strings. Hence, the size of this field is equal to 4 bytes.

- **lifetime:** A *lifetime* is an optional field in an Interest packet, yet a mandatory field when the Interest is propagated within a network. A lifetime is a time counter generated by an application that indicates the time left before the Interest expires. A lifetime is a numerical identifier in milliseconds, that must not be negative. If a lifetime is missing from an Interest, a default lifetime of 4×10^4 milliseconds is set by the forwarder before propagating the Interest upstream. Consequently, and similar to the name field of an Interest packet, the size of a lifetime may vary.
- **hoplimit:** A *hoplimit* is an optional field in an Interest packet, yet a mandatory field when the Interest is propagated within a network. A hoplimit is a non-negative integer within the range $\{0, 2^8 - 1\}$, used to indicate the number of routers that an Interest may traverse before being discarded. In the case that a hoplimit field is missing from an Interest packet received at an NDN router, the router may decide to add a hoplimit field. The size of this field is equal to 1 byte.

2.1.2.3 Data Packet

For the purpose of this thesis, a Data packet is described as follows:

- **name:** A *name* is a mandatory field in a Data packet, which represents the name content identifier of the Data. This field is equivalent to the corresponding field of an Interest packet, as described in Section 2.1.2.2. Nonetheless, and in contrast to an Interest packet where a number of name components is optional, a name in a Data packet is composed of both the *GenericNameComponent* and the *ImplicitSHA256DigestComponent*. An *ImplicitSHA256DigestComponent* is used to uniquely identify a Data packet by attaching the SHA-256 digest of the entire Data packet bits at the end of the name. This convention allows an Interest to

specify the Data packet to be retrieved by attaching the equivalent name component to the name of the Interest packet. The value of this field is represented as a sequence of 64 hexadecimal digits. Hence, the size of this field is equal to 32 bytes.

- **metainfo:** The *metainfo* field is an optional field in a Data packet composed of the *Content Type* and *Final Block* TLVs, used to provide additional information about the Data. The *Content Type* field is used to indicate whether the Data packet contains actual content, a public key or an application-level *Negative Acknowledgment (NACK)*. To distinguish between the different types of Data, a non-negative integer number is used. A non-negative number within the range $\{0, 2^8 - 1\}$ should be sufficient to support the future types of Data. Hence, the size of this field could be considered equal to 1 byte. The *Final Block* field is used to indicate whether a Data packet refers to the last chunk of an object resource, by including the name component of this chunk. If otherwise, the inclusion of this field in a Data packet is optional. Recalling the definition of name components, and the *GenericNameComponent* in particular, a *GenericNameComponent* has no restriction on the type of its value field. Hence, the size of this field may vary.
- **content:** The *content* field is an optional field in a Data packet, that contains the actual content to be retrieved, represented as an arbitrary sequence of bytes.
- **signatureinfo:** The *signatureinfo* field is a mandatory field in a Data packet that combined with the *signaturevalue* field they are used to secure the integrity and/or provenance protection of the content contained within a Data packet. The *signatureinfo* field is composed of a mandatory TLV, called *SignatureType* and an optional TLV, called *KeyLocator*. The *SignatureType* field is used to maintain information about the type of the signature algorithm, e.g. an SHA-256 algorithm.

To distinguish between the different algorithms, a non-negative integer number is used. The value of this number has been defined to be within the range $\{0, 200\}$, where the range $\{5, 200\}$ has been reserved for future assignments. Hence, the size of this field could be considered equal to 1 byte. The KeyLocator field is used to identify the Name of another Data packet that contains a certificate or a public key used to verify the content of this Data packet, or a *KeyDigest* used to identify the public key, assuming that a *trust model* exists [Yu b]. A trust model defines a relationship between name content identifiers and public-private key pairs used by publishers and consumers to sign and verify content. Depending on the type of the signature algorithm used, KeyLocator may be defined as a mandatory, optional or forbidden field. The size of this field is dependent on the type of the signature algorithm, and the length of the private key used in the signing process.

- **signaturevalue:** The signaturevalue field is a mandatory field in a Data packet used to represent the actual bits of a signature. The size of this field is dependent on the type of the signature algorithm, the length of the private key used in the signing process and the fields included in the signatureinfo TLV. At this point it is worth noting, that while the signatureinfo field is included in the calculation of a signature, the signaturevalue field is not. This design convention has been decided to allow a faster signing process over the contents of a Data packet.

For ease of understanding, Fig. 2.1 and 2.2 provide a representation of the Interest and Data packet formats, respectively, following the conventions below:

- i Both the Interest and Data packets have been constructed to include the packet fields necessary and/or related to the context of this thesis.

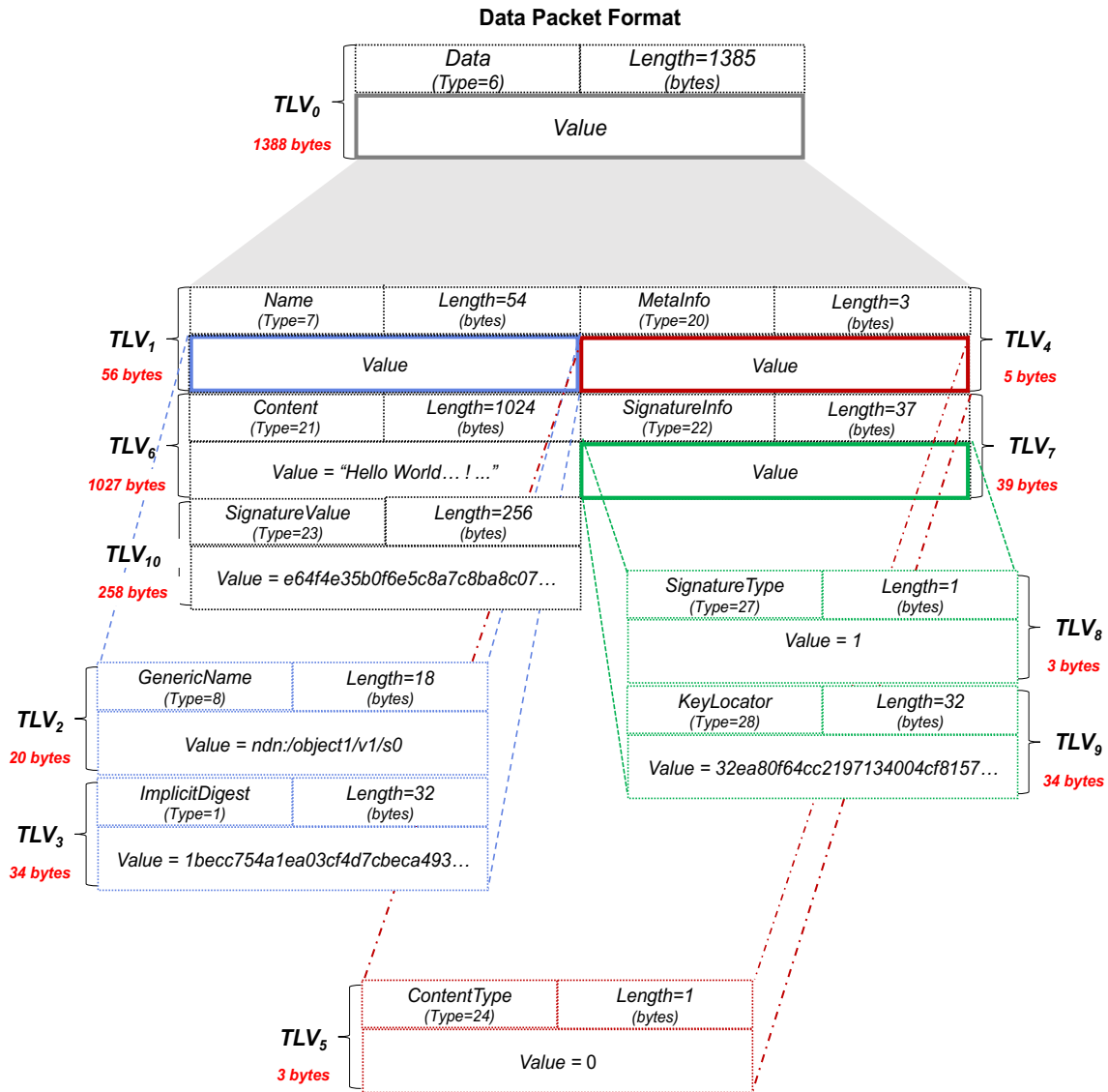


Fig. 2.2: Representation of the fields of a Data packet necessary and/or related to the context of this thesis for the retrieval of the *ndn:/object1/v1/s0* chunk/content.

- ii Both the Interest and Data packets have been constructed to adopt the decimal representation of the fields within a packet, as described in Section 2.1.2.1.
- iii Both the Interest and Data packets have been constructed to use the default and/or the maximum value and/or length suggested for a TLV in the NDN documentation, when possible. These TLVs are the following:
 - *interestlifetime* TLV of the Interest packet, whose value has been set to be equal to the default value suggested in the NDN documentation,
 - *hoplimit* TLV of the Interest packet, whose value has been set to be equal to the maximum value suggested in the NDN documentation,
 - *content* TLV of the Data packet, whose length has been set to be equal to the size of a chunk suggested by the NDN research community [Gill ,Zhou a].
- iv Both the Interest and Data packets have been constructed to refer to the same chunk of an object resource, i.e. *ndn:/object1/v1/s0*, as defined in Section 2.1.1.
- v Besides the number of bytes occupied by the Value field of a TLV, represented by its Length field, both the Interest and Data packets have been constructed to illustrate the total number of bytes of a TLV, including the Type and Length fields. The total size of a TLV is illustrated below the TLV number of a TLV.
- vi The Data packet has been constructed to use arbitrary sequences of hexadecimal digits for the TLVs whose values are determined based on the calculation of a cryptographic algorithm, such as an SHA-256 algorithm, over the contents of a packet. This convention has been made for both simplicity and readability purposes. These TLVs are the: ImplicitSHA256DigestComponent, SignatureValue and KeyLocator

TLVs of a Data packet. To determine the length of the corresponding TLVs, the following convention had been made: a Sha256WithRsa signature algorithm with a 256-length private key has been used to sign the contents of a Data packet.

- vii For ease of readability, the GenericNameComponent field of both the Interest and Data packets, and the ImplicitSHA256DigestComponent field of the Data packet format have been abbreviated as follows: *GenericName* and *ImplicitDigest*.

Both the Interest and Data packet formats presented in this section will be used as a reference point to define any changes necessary on the format of the packets to support the functionality of the mechanisms proposed in the following chapters of this thesis.

2.1.3 Content Store (CS)

A *Content Store (CS)* is a data structure that stores Data on a chunk naming granularity. A CS is searched by name every time an Interest arrives at an NDN router. The time required to complete this search increases the content delivery times experienced by consumers. Since this time is dependent to the size of a CS, the size of a CS needs to be bounded. The recommended size of a CS that stores Data on a chunk naming granularity is between 10GB and 32GB [Arianfar ,Lee ,Perino]. A CS is updated according to a caching policy and a replacement policy. A caching policy is applied to a router upon the arrival of Data. This introduces the problem of on-path caching, according to which a caching decision on content needs to be made. An overview of the challenges that on-path caching raises is presented in Section 1.3.2. A content replacement policy is applied to a router upon the arrival of Data, and if the cache capacity is exhausted. Following the necessity for lightweight operations applied to an NDN router, replacement policies such as the Least-Frequently Used (LFU) and Least-Recently Used (LRU) have been defined

as complex, while shown to provide minor benefits against RaNDom (RND) [Arianfar , Rossi a, Rossini b]. Nonetheless, no numerical results have been presented to define their complexity overhead, while LRU has been adopted in a number of studies, e.g. [Cho , Psaras b, Surlas c]. LRU is the default replacement policy of NDN.

2.1.4 Pending Interest Table (PIT)

A *Pending Interest Table (PIT)* is a data structure that keeps track of the outstanding Interests, i.e. the Interests that have been propagated to a content source(s), but have not been satisfied yet. A PIT entry consists mainly of information extracted from the Interest packet format upon the arrival of an Interest at an NDN router, listed as follows:

- ***name***: A content identifier used to identify a chunk of an object resource, for which an Interest has been received. This information is extracted from the equivalent field of an Interest.
- ***infaces***: A list of incoming faces through which an Interest(s) for this content identifier has been received since the creation of the entry.
- ***outfaces***: A list of outgoing faces through which an Interest(s) for this content has been propagated to a content source(s) since the creation of the entry.
- ***lifetime***: A time counter that once expired triggers the removal of the entry. A lifetime is initialised and updated upon the arrival of an Interest by extracting the value of the equivalent field from the Interest packet format.
- ***nonces***: A list of nonces of the Interests aggregated by the mechanism of request aggregation since the creation of the entry. Nonces are used to detect duplicated

Interests that have been forwarded through multiple faces within a network and discard them. This information is extracted from the equivalent field of an Interest.

A PIT delivers Data to consumer(s) by matching the content identifiers of Data to the faces in the infaces list. By doing so, both Interests and Data are forwarded through the same faces and paths, a process called symmetric forwarding [Yi a, Zhang a]. Symmetric forwarding acts as a native flow-control mechanism in NDN [Wang d, Yi b, Zhou b]. Additionally to this advantage, a number of advantages associated with the structure and the functionality of a PIT have been affirmed, as follows:

- The first advantage refers to the information that may be collected, such as the drop rate of either Interests or Data, the arrival rate of duplicated Interests and the arrival rate of unsolicited Data. Such information may be used by routers to identify communication problems, e.g. the loss of connectivity [Carofiglio b, Yi a].
- The second advantage refers to the request aggregation mechanism. The purpose of this mechanism is to aggregate close-in-time Interests that refer to the same content at an NDN router. Thus, request aggregation is a native flow-control mechanism and a security mechanism of the NDN architecture that aims to reduce the traffic within a network and prevent the expansion of DoS attacks to both consumers and content sources [Ghali].
- The third advantage refers to the identification of unsolicited Data. By identifying unsolicited Data, adversaries that aim to overload the network with malicious information are prevented. This type of DoS attack is called *reflection attack*. Location-based security attacks [Gasti] may also be prevented by allowing the anonymity of both consumers and content sources within a network.

2.1.5 Forwarding Information Base (FIB)

A *Forwarding Information Base (FIB)* is a data structure that keeps track of a list of faces for each advertised content identifier within a network. Since content may be hosted by multiple content sources, a FIB may hold multiple faces for a content identifier. A FIB is populated using a name-based routing service and/or static routes. In concept, a FIB of an NDN router is equivalent to the FIB of the current Internet infrastructure. Nonetheless, a FIB of an NDN router extends the capabilities of a conventional FIB since it is eligible to forward Interests to multiple content sources and qualify the retrieval of Data in parallel. A face is an extended version of an interface in the sense that besides the communication within a network, a face is used to represent the communication between the network infrastructure and an application process at an NDN router.

Similar to the size of a CS and the lookup times, the size of a FIB is dependent on the catalog size. To reduce the size of a FIB a prefix may be used to aggregate alike content identifiers. As an example, the FIB entry *ndn:/object1/v1 - face₁* may replace the two following entries: *ndn:/object1/v1/s0 - face₁* and *ndn:/object1/v1/s1 - face₁*.

2.1.6 Upstream Process

Following the representation of the Interest and Data packet formats presented in Fig. 2.1 and 2.2, Fig. 2.3 provides a representation of the upstream process at an NDN router, described as follows: Upon the arrival of an Interest a router will perform a CS lookup (1). If a CS entry that matches the content identifier of the Interest exists, the content will be propagated to the consumer through the face at which the Interest arrived (2), i.e. *face₂*. If otherwise, a router will perform a PIT lookup (3). If a PIT entry that matches the content identifier of the Interest does not exist, a PIT entry will be created

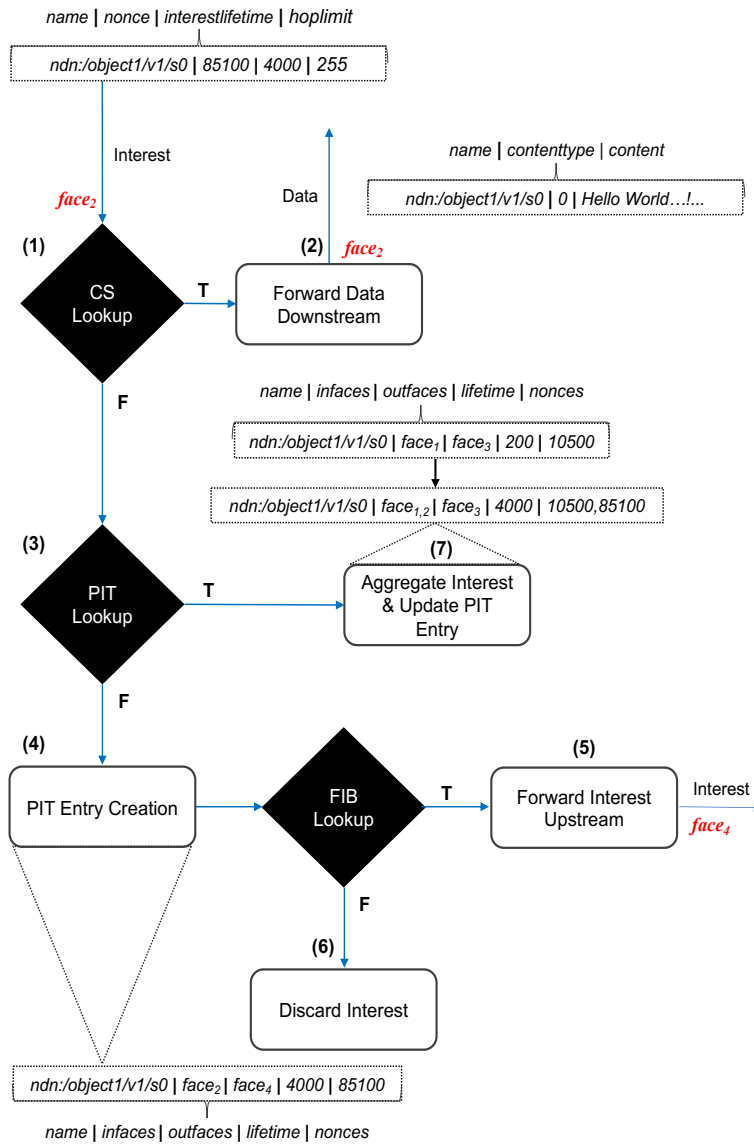


Fig. 2.3: Upstream forwarding process of an Interest at an NDN router, i.e. from a consumer to a content source(s).

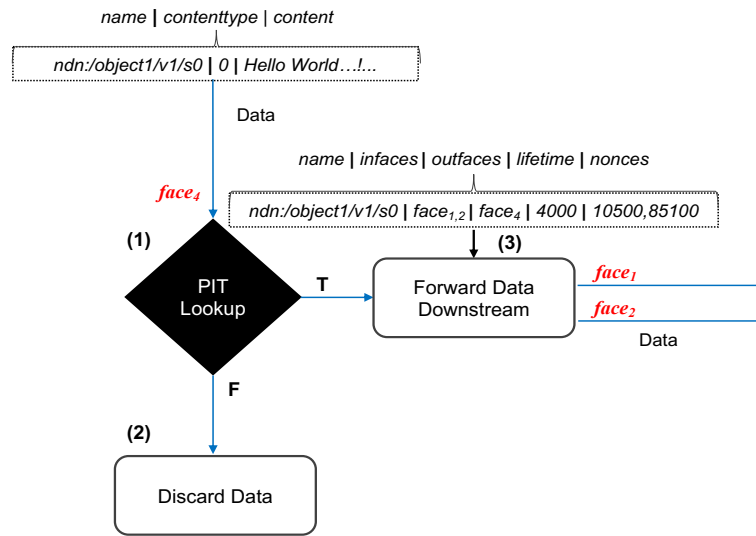


Fig. 2.4: Downstream forwarding process of Data at an NDN router, i.e. from a content source, or another NDN router that possesses a replica, to a consumer(s).

(4). Once a PIT entry is created, the Interest will be propagated upstream to a content source(s) according to the information in the FIB (5), i.e. *face₄*. If a FIB entry that matches the content identifier of the Interest does not exist, the Interest will be discarded (6). If a PIT entry that matches the content identifier of the Interest exists, the Interest will be aggregated. The existing PIT entry will be updated according to the information included in the incoming Interest, such as its lifetime and nonce (7). The incoming face on which the Interest has been received will be added to the list of infaces.

2.1.7 Downstream Process

Following the representation of the upstream process in Fig. 2.3, Fig. 2.4 provides a representation of the downstream process at an NDN router, described as follows: Upon the arrival of Data a router will perform a PIT lookup (1). If a PIT entry that matches the content identifier of the Interest does not exist, the packet is unsolicited and it will

be discarded (2). If otherwise, a replica of the Data will be propagated to each of the consumers interested in retrieving this content through the faces in the infaces list (3), i.e. $face_1$ and $face_2$. After the propagation of Data, the PIT entry will be deleted.

2.2 On-path Caching: Categorisation

On-path caching aims to reduce the amount of traffic propagated within a network and beyond the boundaries of this network to the Internet infrastructure by exploiting the caching capacity of routers on delivery paths during the downstream process. For this purpose, on path caching is integrated with the forwarding mechanism that commonly follows a chunk naming granularity. Due to this integration, on-path caching is limited by a number of factors, such as the cache size that the routers on delivery paths are bound to follow to maintain acceptable content delivery times. On-path caching is also affected by the dynamic environment of ICN architectures [Ioannou , Katsaros]. A description of the challenges in on-path caching is presented in Section 1.3.1.

The aforementioned dependencies constitute operations such as monitoring, collection of statistical information, and advertisement of the cached content into a name resolution service and/or a name-based routing service impractical [Ahlgren a, Rossi a]. Due to these limitations, on-path caching is a short-term decision where the use of lightweight and heuristic techniques is preferable to the use of complex techniques.

Approaches to on-path caching may be categorised in relation to the criteria below:

- **Naming granularity:** The naming granularity to which a caching policy is applied may be different between ICN architectures. Based on this criterion caching policies may be categorised into *chunk-based* or *object-based* caching policies, respectively. NDN has been defined to cache content on a chunk naming granularity.

In theory, caching policies can be applied either on a chunk or an object naming granularity with minor or no differences (Section 2.2.4). Hence, their performance is expected to be equivalent regardless of the level of naming granularity. The main difference between chunk-based and object-based caching policies is that the former enables the retrieval of individual and/or specific chunks of an object resource.

- **Caching criterion:** Caching policies may be based on information with regard to the location of routers on delivery paths or the location of routers within the boundaries of an ISP/AS network and the popularity of content. Based on these criteria caching policies may be categorised into *location-based* and *popularity-based*, respectively. Caching policies may also be based on multiple types of information bound together to construct a probability. Such caching policies are called *probability-based* caching policies. A categorisation of the caching policies reviewed in this thesis is presented in the following sections.

- **Scale:** Caching policies may be applied either on the scale of the Internet infrastructure or an ISP/AS network [Jacobson , Koponen]. Based on this criterion caching policies may be categorised into *Internet-scale* and *ISP/AS-scale* caching policies, respectively. In practice, an Internet-scale caching policy strongly depends on the business models established between ISP/ASes [Chang , Cheriton , Dhamdhere a]. Since on-path caching may be applied to routers regardless of the underlying network topology (Section 1.3.1), enforcing implicit at minimum, cooperation between the routers on a delivery path, implicit cooperation should also be supported by the ISP/ASes within the Internet infrastructure. In addition to this, ISP/ASes maintain connectivity to their siblings by paying a fee to the transit ISP/ASes to propagate traffic. Thus, implicit cooperation between ISP/ASes may

only exist if each ISP/AS benefits from it [Ma , Wang c]. Since the benefit of ISP/ASes that perform on-path caching within the Internet infrastructure is yet to be decided [Kamiyama , Pacifici , Pham , Rajahalme], for the rest of this thesis, all caching policies are assumed to operate within an ISP/AS network.

- **Cooperation:** Caching policies may be based on information collected at the router performing the caching decision, or information collected either implicitly or explicitly by a group of routers within a network. Based on this criterion caching policies may be categorised into *autonomous*, *implicitly-cooperative*, and *explicitly-cooperative* caching policies, respectively. An example of implicitly collected information is the information injected into the content request and content reply packet formats, i.e. the Interest and Data packets, such as the hop-count metric. An example of explicitly collected information is the exchange of messages between neighbour routers to inform each other about the content that they possess, an operation that allows them to eliminate content redundancy, yet increase the network traffic. In the former form of cooperation, two essential assumptions are made:

- ◊ No information, other than the information required to guarantee the functionality of a caching policy, is propagated through the network infrastructure.
- ◊ If extra information is propagated, it is propagated only between a centralised configuration module that maintains full knowledge of the network topology [Anand , Rossi b, Saino a] and the routers within a network.

Cooperative caching policies may also be categorised depending on the scale of the network in which they operate, into *global-cooperative* and *local-cooperative* caching policies. The first type of cooperation refers to all routers within a network. The

second type of cooperation refers to a group of routers within a network, such as the routers on delivery paths or the neighbour routers of a router. According to this criterion, and considering the communication overhead that global-cooperative caching policies are expected to introduce, only local-cooperative caching policies refer to on-path caching [Li b, Li e, Ming , Sourlas b, Sourlas a, Wang b].

2.2.1 Location-based Caching

Initial caching policies have been based on the criterion of the location of routers on delivery paths. This way caching policies aim to adjust to the dynamics of a network. For ease of reference, the term *location-based caching* is used to refer to the caching policies that utilise the location criterion. Well known examples of location-based caching policies are the *Leave Copy Last (LCL)* [Sourlas d] and *Leave Copy Down (LCD)* [Rossi a].

LCL is one of the first caching policies proposed in the context of on-path caching. LCL aims to bring content closer to consumers by caching it at a fixed set of routers, i.e. the last router on a delivery path connected to consumers, called *edge router*. By doing so, LCL aims to reduce the number of hops traversed to reach a content source(s), thus reduce both the network traffic caused due to the propagation of content requests and content replies and the content delivery times experienced by consumers.

A generalised approach to LCL is LCD. LCD progressively caches content one hop closer to consumers each time that a content request arrives on a forwarding path. This way LCD aims to maintain popular content on delivery paths. Nonetheless, LCD caches unpopular content since no distinction on the content is made, while it increases the content redundancy on delivery paths. To prevent this increase, a modification of LCD, called *Move Copy Down (MCD)* [Laoutaris], has been proposed that deletes the existing replica of content once the content is progressively cached one hop closer to consumers.

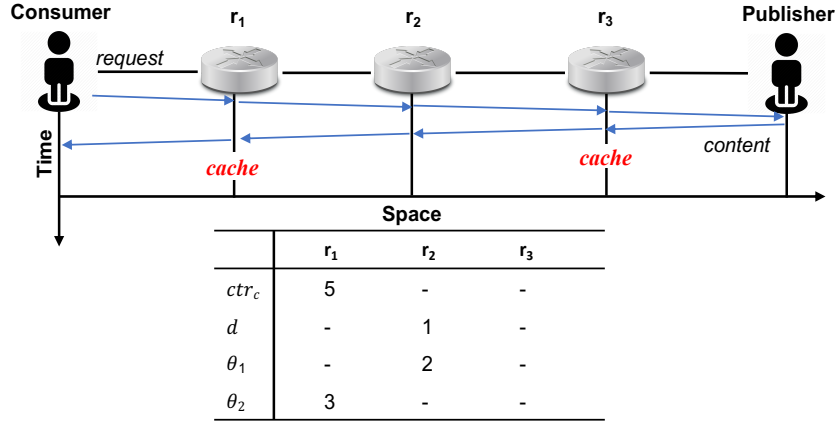


Fig. 2.5: Representation of the PCP caching policy, where r_3 caches content according to LCD, r_2 caches content if the number of faces where the requests arrived $d \geq \theta_1$ and r_1 if the popularity counter for this content $ctr_c \geq \theta_2$; θ_1 and θ_2 are constants.

LCD is based on the assumption that the router that serves a content request may indicate to its neighbour router on a delivery path to cache content based on either a flag or a hop-count metric injected into the content reply packet format.

LCD has been used in the construction of the *Progressive Caching Policy (PCP)* [Wang a]. PCP aims to cache popular content using thresholds. Such caching policies are described in Section 2.2.2.1. According to the operation of PCP an *immediate router*, i.e. the first router on a delivery path will cache content according to LCD, an *edge router* will cache content if the number of content requests for this content ctr_c is $ctr_c \geq \theta_2$, an *intermediate router* that is neither an immediate nor an edge router will cache content if the number of faces where the content requests for this content d arrived is $d \geq \theta_1$. Besides the usability of θ_1 and θ_2 , no information on their selection is given.

The operation of PCP is presented in Fig. 2.5, where router r_3 will cache the content according to LCD, while router r_1 will cache the content since its popularity counter

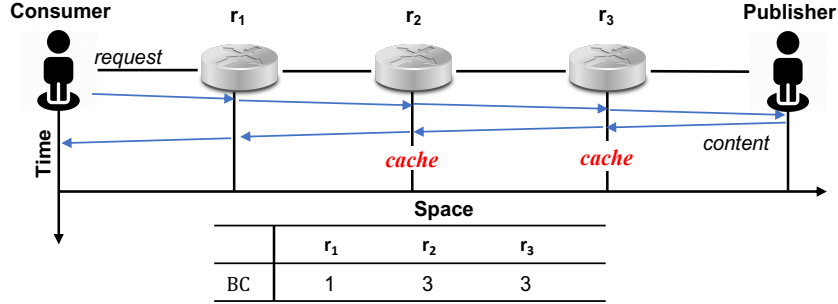


Fig. 2.6: Representation of the HVC caching policy, where content is cached at the routers with the highest BC value.

$ctr_c \geq \theta_2$. Router r_2 will not cache the content since $d < \theta_1$.

Besides the location criterion of routers on delivery paths, caching policies have based their caching decision on the location of routers within the boundaries of an ISP/AS network by utilising centrality metrics such as the *Betweenness Centrality (BC)*. BC indicates the frequency that router r is included in the sets of shortest paths within a network between all pairs of routers: $BC_r = \sum_{r_x, r_y \neq r}^V dst(r_x, r_y, r) / dst(r_x, r_y)$, where $dst(r_x, r_y, r)$ is the shortest-path distance between routers r_x and r_y through r .

BC has been used to cache content at a fixed set of routers, i.e. the routers on delivery paths with the highest centrality value [Chai]. For this purpose, the highest BC value of the routers that have been traversed during the propagation of a content request to a content source is injected into the content request's packet format. Upon content delivery, this value is injected into the content reply's packet format. For the rest of this thesis, the term *Highest-Value Caching (HVC)* is used to refer to this caching policy. Besides the BC metric, any centrality metric may be used instead.

The operation of HVC is presented in Fig. 2.6, where the BC metric for routers r_1 , r_2 and r_3 is equal to 1, 3 and 3, respectively. The highest BC value of the routers along

the forwarding path is equal to 3. Consequently, both routers r_2 and r_3 whose BC value is equal to the highest centrality value will cache the content.

Caching policies that ground their operation on centrality metrics require the existence of a *Network Manager Module (NMM)*. An NMM is expected to be aware of the network topology and to calculate and assign centrality values to routers based on Shortest Path Routing (SPR); alternative routing protocols may be used instead. Besides the requirement of an NMM, the performance of caching policies based on centrality metrics may be affected by both the network topology and its connectivity. Finally, all caching policies based on centrality metrics neglect content popularity.

2.2.2 Popularity-based Caching

Besides the location criterion of routers on delivery paths or within the boundaries of an ISP/AS network, the criterion of content popularity has been used to perform a caching decision. This way, caching policies aim to adjust to the dynamics of the network and to the network traffic patterns. For ease of reference, the term *popularity-based caching* is used to refer to the caching policies that utilise the content popularity criterion.

Popularity-based caching policies aim to cache popular content and discard unpopular content. This way, popularity-based caching policies aim to prevent the problem of *cache pollution* caused by caching *one-timer* objects at routers. The term cache pollution refers to the utilisation of caches by unpopular content, that prevent caching popular content. The term one-timer objects refers to object resources that are requested once. One-timer objects are estimated between 45-75% of the object requests [Mahanti], which caching policies based on alternative caching criteria, such as the location of routers, fail to identify. The cache pollution problem has been shown to increase both the network traffic and the content delivery times experienced by consumers [Janaszka ,Li b].

Popularity-based caching policies may base their caching decision on popularity information collected either at the scale of delivery paths or at the local router that performs the caching decision, called *path-scale popularity* and *local-scale popularity*, respectively. In the first category, caching is applied to at least one router on a delivery path upon the arrival of the first content request. Consequently, path-scale popularity fails to address the cache pollution problem effectively. In the second category, caching is applied to a router based on the content popularity observed at this router. Examples of caching policies based on the path-scale popularity criterion are the LCD and MCD, described in Section 2.2.1. Therefore, the focus of this section lies on local-scale popularity.

Popularity-based caching policies may be further categorised into *threshold-based* and *correlativity-based* caching policies based on the method used to define local-scale popularity, i.e. a threshold thr applied to the *popularity counter* ctr_c of a content or a comparison between the popularity counters' values of multiple contents.

2.2.2.1 Threshold-based Caching

Threshold-based caching is based on the use of a threshold thr applied to the popularity counter ctr_c of a content to determine whether the content is popular. If $ctr_c \geq thr$, the content is considered to be popular. If $ctr_c < thr$, the content is considered to be unpopular [Domingues, Li b, Wang a]. Unpopular content is discarded. In its simplest form, a popularity counter ctr_c is additively increased upon the arrival of a content request and decreased periodically by rate γ to ensure the eviction of content from caches. γ is assumed to be set a priori and to remain fixed. For this purpose the content demand within a network is also assumed to be known a priori and to remain fixed [Domingues]. Popular contents should comport with a higher rate γ .

Due to the dynamic environment of ICN, and its dependency on the traffic patterns of

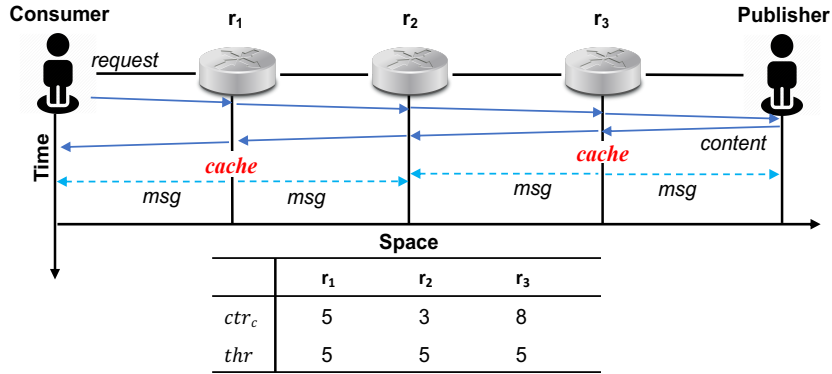


Fig. 2.7: Representation of the MPC caching policy, where content is cached at routers if the popularity counter of the content ctr_c is higher or equal than a threshold thr .

a network and content demand, the definition of a threshold can be challenging, resulting in outdated calculations and unutilised cache capacity [Bernardini , Janaszka , Li b]. Examples of threshold-based caching policies are the PCP and *Most Popular Content (MPC)* [Bernardini]. PCP has been described in Section 2.2.1. MPC aims to distribute popular content to all neighbours of a router besides the delivery paths. For this purpose, MPC distributes an extended content reply packet, called *suggestion message*, to notify its neighbours. Upon the arrival of a suggestion message a router will decide whether to cache content according to its local information. Depending on the connectivity of the network, the propagation of suggestion messages - or replicas - of the cached content is expected to significantly increase the network traffic.

The operation of MPC is presented in Fig. 2.7, where the popularity counters ctr_c for routers r_1, r_2 and r_3 are equal to 5, 3 and 8, respectively. The threshold thr is set to be equal to 5 for all routers. Since the threshold of both r_1 and r_3 is equal or higher to the threshold defined, both routers will cache the content while suggestion messages will be propagated to their neighbours. In this case, r_2 that is the intermediate router

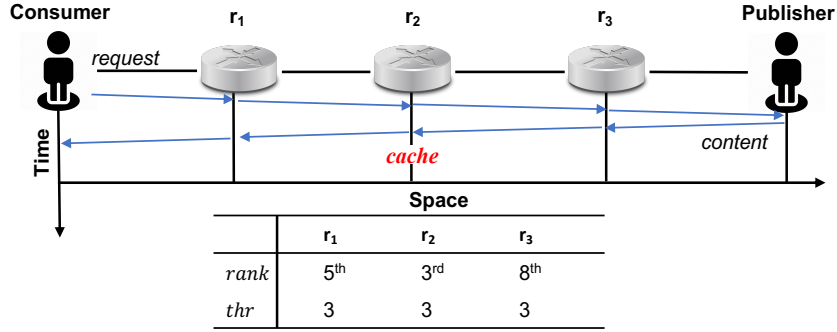


Fig. 2.8: Representation of the PbLRU caching policy, where content is cached at routers if the rank value of the content is within the limits defined by the threshold thr .

between r_1 and r_3 will receive the same suggestion message twice.

2.2.2.2 Correlativity-based Caching

Correlativity-based caching is based on the comparison of the popularity counters' values between contents to determine their correlation, usually during a time interval ΔT . The definition of a time interval able to capture the dynamics of a network is dependent to a number of factors including the network topology, the number of consumers and the content popularity distribution. The collection and the analysis involved into this large set of information result into a complicated system that may only be maintained by an NMM, an ISP, or a similar entity [Ahmed , Domingues , Sourlas a]. For this reason, a common technique used is to define an infinite ΔT [Badov , Cho , Xie].

Besides the probability-based caching policies described in the following section, that may utilise correlativity-based popularity calculations, *Popularity-based LRU (PbLRU)* [Janaszka] is the only correlativity-based caching policy proposed within the context of on-path caching. PbLRU is based on the execution of an Exponential Moving Average (EMA), used to update the popularity of content as follows $pop_{\Delta T'} = pop_{\Delta T} \times (1 - \alpha) +$

$\alpha \times ctr_c$, and the LRU replacement policy. The popularity counter ctr_c is increasingly updated upon the arrival of a content request. PbLRU decides whether to cache content by comparing $pop_{\Delta T'}$ against a threshold thr . The use of a threshold in PbLRU is different to threshold-based caching, described as follows: A content is defined to be popular and is cached if it is located to a position above the one indicated by the value of the threshold. For this purpose, all contents are sorted in a decreasingly ordered list according to their $pop_{\Delta T'}$ values. If otherwise, the content is discarded. According to the authors, the construction of a decreasingly ordered list may impose a significant computational overhead at routers. Even though this overhead has not been explicitly specified, known sorting algorithms such as Quicksort and Mergesort can provide an estimation of the complexity involved, i.e. $\theta(n \times \log(n))$. A sorting algorithm in PbLRU is expected to be executed upon the arrival of a content request.

The operation of PbLRU is presented in Fig. 2.8, where the order of the content in the decreasingly ordered list at routers r_1, r_2 and r_3 is equal to 5, 3 and 8, respectively. The threshold thr is set to be equal to 3 for all routers, meaning that the content will be cached at a router if its order within the decreasingly ordered list is either 1st, 2nd or 3rd. Since the order of the content at routers r_1 and r_3 is lower than the order of the content defined by the threshold, the content will be cached at router r_2 alone.

2.2.3 Probability-based Caching

Besides the location criterion of routers on delivery paths or within the boundaries of an ISP/AS network and the criterion of content popularity, a fixed probability defined a priori [Arianfar ,Laoutaris], or a combination of criteria used to construct a dynamic probability [Psaras b,Wang e], have also been used to perform a caching decision. For ease of reference, the terms *fixed-probability caching* and *dynamic-probability caching* are

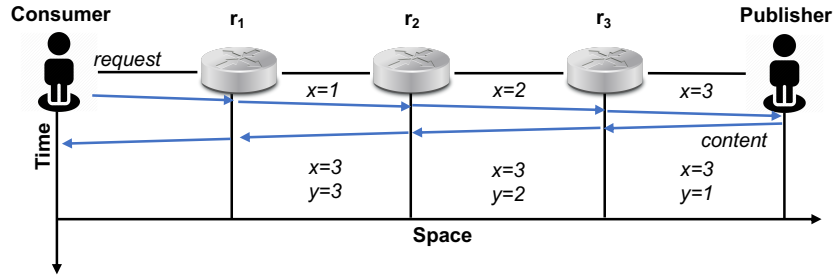


Fig. 2.9: Calculation of the number of hops traversed on a forwarding path by a content request and the number of hops traversed on a delivery path by a content reply, x and y , respectively, used in dynamic-probability calculations.

used to refer to each category of caching policies, respectively.

2.2.3.1 Fixed-probability Caching

Fixed-probability caching is based on the use of a fixed probability p defined a priori. Fixed-probability caching is easy to deploy and involves no additional information that needs to be collected or exchanged between routers. Nonetheless, fixed-probability caching is unable to exploit any knowledge with regard to either the location of routers or the content popularity, resulting in high resource consumption and content redundancy.

Leave Copy Everywhere (LCE) is the first caching policy proposed in the context of on-path caching. LCE is a fixed-probability caching policy with $p=1$, that caches content at every router on a delivery path. LCE has been shown to result in high resource consumption and content redundancy compared to the alternatives [Che , Laoutaris , Wong], yet faster content dissemination [Bernardini , Chai , Cho].

A generalized approach to LCE is *RaNDom (RND)* where $p \in [0, 1]$ [Psaras b, Xiaoyan]. Due to its random factor, RND has been shown to favour popular content, while its performance is dependent to the value of probability p [Arianfar].

2.2.3.2 Dynamic-probability Caching

Dynamic-probability caching is based on the use of multiple criteria combined together to construct a probability p . Dynamic-probability caching policies are more complex compared to fixed-probability caching policies or caching policies based on a single caching criterion. Nonetheless, dynamic-probability caching policies are also expected to make better caching decisions that yield a higher performance gain.

2.2.3.2.1 ProbCache (PC)

ProbCache (PC) is the first caching policy proposed in the context of dynamic-probability caching, composed of the *TimesIn* and the *CacheWeight* factors, formulated as follows: $p = PC = TimesIn \times CacheWeight$ [Psaras b]. PC decides to cache content based on the cache capacity of delivery paths defined by the *TimesIn* factor, and the *CacheWeight* factor that aims to cache content closer to consumers. For this purpose, *CacheWeight* is used to define the location of routers on delivery paths. The rationale behind this choice is that content that is destined to consumers that are further away from a content source should be less likely to be cached at routers located closer to the content source. This allows the consumers that are closer to a content source to utilise the cache capacity of intermediate routers. Following this design convention, PC aims to fairly allocate the cache capacity of delivery paths between contents.

$TimesIn = (\sum_{r=1}^{x-y+1} N_r) / (T_{tw} \times N_x)$ and $CacheWeight = y/x$. N_r is the time for which router r should cache content determined by the router's cache size. T_{tw} is the time for which content should be cached on delivery paths. According to the authors, T_{tw} depends on both the network topology and the network traffic, thus it should be defined a priori by an ISP. N_x is the average cache size of routers on delivery paths. Given that the cache size of the remaining routers on delivery paths is unknown to

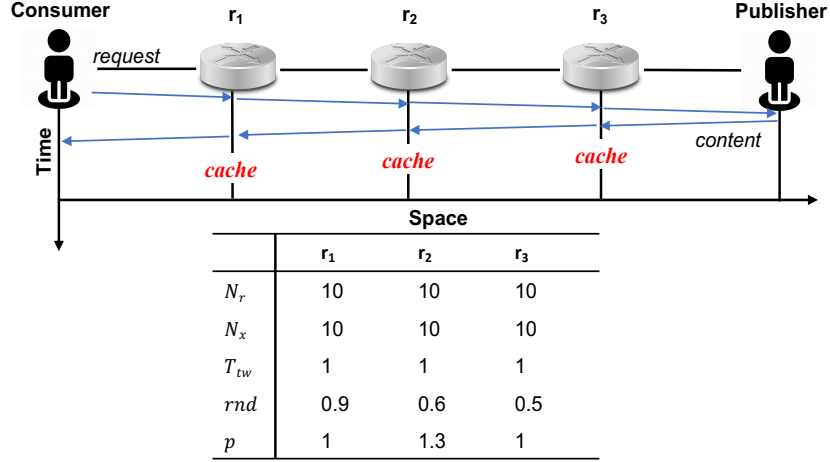


Fig. 2.10: Representation of the PC caching policy, where content is cached at routers if the probability p calculated utilising the N_r , N_x and T_{tw} parameters is higher or equal than a random value rnd ; N_r , N_x and T_{tw} are related to the cache capacity of routers.

the router that performs the caching decision r , every router assumes an average cache size equal to its own. x is the number of hops traversed from a consumer to a content source. y is the number of hops traversed from a content source to router r . A content source may be either a publisher or a router that possesses a replica. Both x and y are updated at every router on the forwarding and the delivery paths. For this purpose, both parameters are injected into the content request and content reply packet formats. NDN follows a symmetric forwarding strategy. Consequently, CacheWeight $\in [0, 1]$.

The operation of PC is presented in Fig. 2.10, where both N_x and N_r are set to be equal to 10 for all routers and T_{tw} is set equal to 1. x and y are set according to Fig. 2.9. The values of the random variable rnd to be compared against the probability value p are set to 0.9, 0.6 and 0.5 for each router, respectively. Since the probability at each router $p > rnd$, the content will be cached at all routers.

In contrast to its initial purpose, PC has been shown to result in an unfair allocation

of the cache capacity since the probability p at the routers on delivery paths does not increase proportionally to the distance of a router from the content source [Psaras a]. In addition to this, PC may result in probability values higher than 1. To ensure that the caching probability is proportional to the distance of a router from a content source and that $p \in [0, 1]$ *ProbCache+* (*PC+*) has been proposed, where the *CacheWeight* = $(y/x)^x$.

2.2.3.2.2 Hop-based Probabilistic Caching (HPC)

Hop-based Probabilistic Caching (HPC) is a dynamic-probability caching policy composed of the *CacheWeight_y* and the *CacheWeight_{RT}* factors, formulated as follows: $p = HPC = CacheWeight_y \times CacheWeight_{RT}$ [Wang e]. HPC aims to decrease content redundancy at the routers on delivery paths by progressively caching content closer to consumers using the *CacheWeight_y* factor at the routers that are most likely to maintain the content, determined by the *CacheWeight_{RT}* factor.

$CacheWeight_y = 1/(y + \alpha)$, where similar to PC y is the number of hops traversed from a content source to the router that performs caching. y is updated at every router on the delivery paths. For this purpose, y is injected into the content reply packet formats. α is a constant integer dependent to the network cache capacity; no further information on the calculation of α is given. To this end, HPC requires the existence of an NMM. For evaluation purposes, α is set equal to 1. $CacheWeight_{RT} = RT_m/RT_{exp}$, if $RT_m < RT_{exp}$, and 1 if otherwise. RT_m is the mean residence time of contents at routers, while RT_{exp} is the expected mean residence time of content at routers. RT_{exp} is defined a priori by an ISP, depending on the traffic load of a router. According to the authors, HPC is expected to yield its highest performance gain when $RT_{exp} = RT_m$.

According to the definition of *CacheWeight_y*, its value decreases as a content reply is propagated to a consumer. This contradicts the aim of HPC to progressively cache

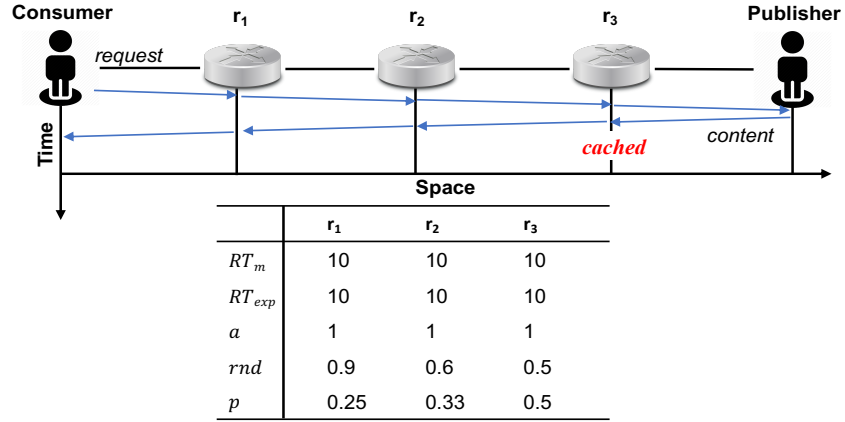


Fig. 2.11: Representation of the HPC caching policy, where content is cached at routers if the probability p calculated utilising the RT_{exp} and RT_m parameters and the α constant is higher or equal than a random value rnd ; RT_{exp} and RT_m are related to the residence time of content at routers.

content closer to consumers. In addition to this, the value of $CacheWeight_y$ is indistinguishable for contents originated from the same content source and destined to different consumers. Last, according to the formulation of $CacheWeight_{RT}$, HPC depends on the mean values of RT_m and RT_{exp} . Mean values are expected to result in either an underestimation or an overestimation of the actual central value if the sample distribution is asymmetrical. Given the fact that the residence time of contents at routers depends on the popularity distribution and that the popularity distribution is a skewed distribution [Pentikousis], mean values are not a representative metric.

The operation of HPC is presented in Fig. 2.11, where both RT_{exp} and RT_m are set to be equal to 10 for all routers and α is set equal to 1. y is set according to Fig. 2.9. The values of the random variable rnd to be compared against the probability value p are set to 0.9, 0.6 and 0.5 for each router, respectively. Since the probability at router

r_3 is the only probability for which $p \geq rnd$, only router r_3 will cache the content.

2.2.3.2.3 Congestion-Aware Caching (CAC)

Congestion-Aware Caching (CAC) [Badov] is a dynamic-probability caching policy composed of the *Popularity* (P_f) and *Download* (Dwn_f) factors, formulated as follows: $p = CAC = P_f \times Dwn_f$. CAC aims to reduce the content delivery times of consumers by caching popular content defined by the P_f factor at the downstream routers of congested links defined by the Dwn_f factor, where Dwn_f indicates the time saved by consumers to retrieve content. Similar to the *CacheWeight* factor of PC/PC+ and the *CacheWeight_y* factor of HPC, the Dwn_f factor is used to define the location of routers on delivery paths.

$P_f = ctr_o / \sum_{\forall o} ctr_o$, where ctr_o is the popularity counter of object o , equal to the number of object requests received for this object and $\sum_{\forall o} ctr_o$ is the sum of object requests for all objects. $Dwn_f = S_o / \min(BW_{Up} \cup BW_{Dwn}) - S_o / \min(BW_{Dwn})$, defined as the difference between the fraction of the size of object o , S_o to the minimum bandwidth of the upstream and downstream links, i.e. $\min(BW_{up} \cup BW_{down})$, and the fraction of S_o to the minimum bandwidth of the downstream links for the given flow. Both $\min(BW_{up} \cup BW_{down})$ and $\min(BW_{down})$ are updated at every router on the forwarding and delivery paths, while S_o should be known a priori. For this purpose, both parameters are injected into the content request and content reply packet formats.

CAC is based on the assumption that the bandwidth of a link is fairly allocated between objects. Nonetheless this prerequisite does not hold true since different objects may require a different amount of bandwidth. Moreover, the estimation of the available bandwidth heavily depends on the network traffic patterns. Consequently, and according to the authors, such parameters are easier to become stale and to lead to suboptimal

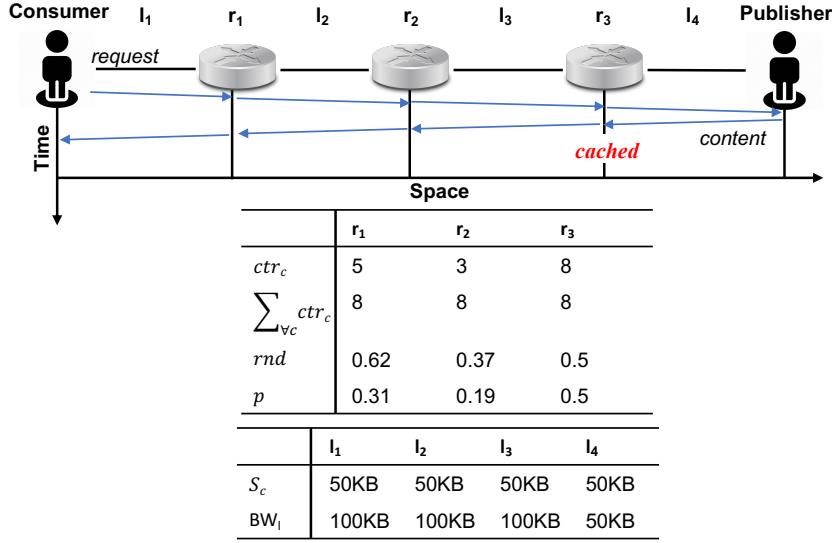


Fig. 2.12: Representation of the CAC caching policy, where content is cached at routers if the probability p calculated utilising the ctr_o , $\sum_{\forall o} ctr_o$, S_o and BW_l parameters is higher or equal than a random value rnd ; ctr_o and $\sum_{\forall o} ctr_o$ are related to the popularity of content, while S_o and BW_l are related to the bandwidth consumption.

caching decisions. Moreover, when $BW_{Dwn} < BW_{Up}$, the gain of caching content at the downstream end routers is dependent on the P_f factor alone.

The operation of CAC is presented in Fig. 2.12, where the popularity counters ctr_o for routers r_1 , r_2 and r_3 are equal to 5, 3 and 8, respectively, while the sum of content requests $\sum_{\forall o} ctr_o$ is equal to 8 for all routers. The size of content S_o is set to be equal to 50KB for all routers. The BW of the link between router r_3 and the publisher is set to be equal to 50KB, while every other link is set to be equal to 100KB. The values of the random variable rnd to be compared against the probability value p are set to 0.9, 0.6 and 0.5 for each router. Since the probability at router r_3 is the only probability for which $p \geq rnd$, only router r_3 will cache the content.

Name Abbreviation	Naming Granularity	Cooperation	Implicit Cooperation
LCL	Object &/OR Chunk	Implicit	Packet-format
MCD	Object &/OR Chunk	Implicit	Packet-format
PCP	Object &/OR Chunk	Implicit	Packet-format
HVC	Object &/OR Chunk	Implicit	Packet-format & Centralised
MPC	Object &/OR Chunk	Autonomous	-
PbLRU	Object &/OR Chunk	Autonomous	-
LCE	Object &/OR Chunk	Autonomous	-
RND	Object &/OR Chunk	Autonomous	-
PC+	Object &/OR Chunk	Implicit	Packet-format
HPC	Object &/OR Chunk	Implicit	Packet-format
CAC	Object &/OR Chunk	Implicit	Packet-format

Table 2.1: Summary of the caching policies reviewed in this thesis, categorised with regard to the caching level and type of cooperation criteria.

2.2.4 The Evolution

In this section, a taxonomy of the caching policies reviewed in this thesis is given according to the categorisation criteria defined in Section 2.2. More specifically, Table 2.1 provides a summary of the naming granularity and the type of cooperation of the caching policies. The network scale criterion has been omitted since all caching policies are assumed to operate within an ISP/AS network. Besides this criterion, the PC and LCD caching policies have also been omitted, since they share the same criteria with PC+ and MCD, respectively. According to Table 2.1, a number of observations may be drawn:

- The first observation refers to the naming granularity of a caching policy. All caching policies may be applied either on an object or a chunk naming granularity. This conclusion has been based on the functionality of a caching policy and besides the naming granularity to which caching policies have been proposed to operate.

- The second observation refers to the type of cooperation of the router that performs a caching decision and the rest of the routers within a network to collect the information necessary for its operation. Based on this criterion, caching policies may be defined as: autonomous, if the information is collected locally at a router, and implicitly or explicitly-cooperative if the information is collected by cooperating either implicitly or explicitly with the rest of the routers within a network. The majority of caching policies follow an implicit cooperation by injecting the information necessary into the content request and/or content reply packet formats. This is due to the fact that cooperative caching policies are expected to outperform autonomous caching policies, yet explicitly-cooperative caching policies are expected to introduce extra communication overhead within a network [Li d, Wang b]. Recalling Section 2.2, the term *centralised* is used to refer to the information propagated between a centralised configuration module that maintains full knowledge of the network topology, such as an NMM, and the routers within a network. In some cases, both types of cooperation may co-exist, e.g. HVC.

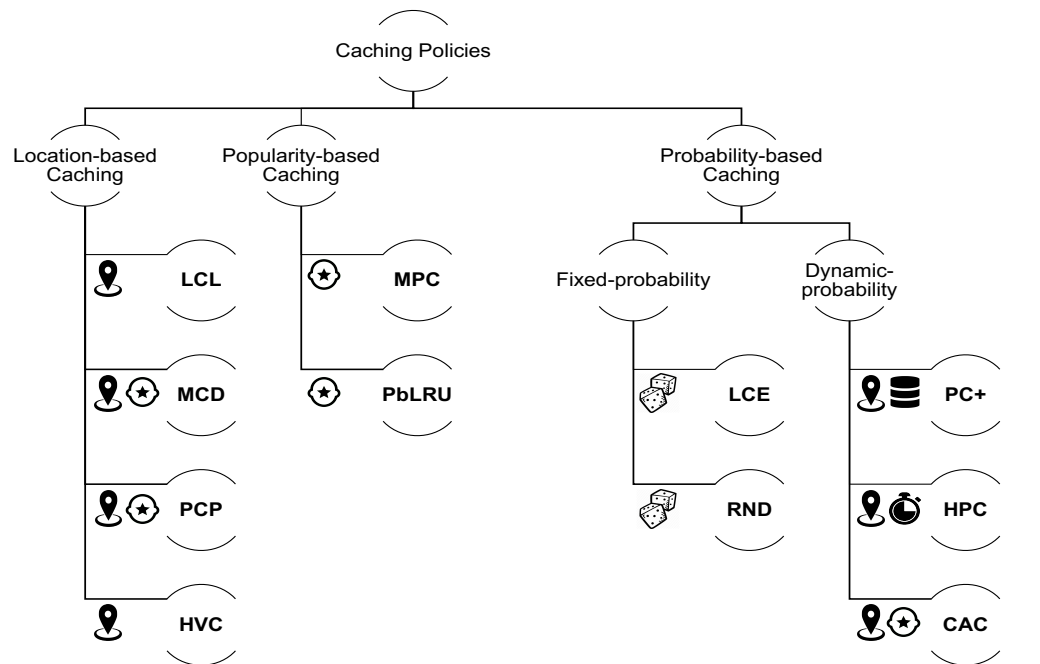
Fig. 2.13 provides a taxonomy of the caching policies according to the caching criteria used; each criterion is explicitly described in the table placed at the bottom of the figure. This information will ease the identification of potential trends on the evolution of on-path caching policies. Similar to Table 2.1, both PC and LCD have been omitted. Consulting Fig. 2.13, a number of observations may be drawn, summarised as follows:

- The first observation refers to the evolution of the caching policies with regard to the caching criteria used. Caching policies have evolved from initial implementations that base their caching decision on a simple criterion, such as the edge routers on delivery paths or a random probability, into implementations that consider a

combination of advanced network-related and content-based criteria. By incorporating more advanced criteria into the caching decision process, caching policies aim to adjust to the dynamics of the network and the network traffic patterns, hence provide a higher performance gain. Examples of such criteria are the distance of a router from a consumer and/or a content source defined in hop-counts, which determines the network traffic introduced within a network, the residence time of content at routers on delivery paths and the content popularity.

- The second observation refers to the criterion of the location of a router and the criterion of content popularity, both of which have been broadly used within the context of on-path caching. At this point it is worth noting, that either caching criterion may be incorporated into the caching decision process in a number of ways. As an example, the location of routers on delivery paths may be a fixed location, e.g. the edge router on a delivery path, or a location determined dynamically, based on a probability constructed by the number of hops traversed by the content request and/or the content reply packets on the forwarding and/or the delivery paths. Moreover, content popularity may be applied on either a path-scale or a local-scale, while local-scale popularity may be further categorised into threshold-based and correlativity-based popularity. Besides the way that a caching criterion may be adopted into the caching decision process, the aforementioned taxonomy highlights the importance of either caching criterion in on-path caching.

The conclusions derived from this section with regard to the evolution of on-path caching policies and the caching criteria used will help the researcher to construct the dynamic-probability caching policy proposed in chapter 3.








	Router Location 	Content Popularity 	Fixed Probability 	Residence Time 	Cache Capacity 
LCL	fixed: edge routers on delivery paths	-	-	-	-
MCD	fixed: routers connected to the content source on delivery paths	path-based popularity	-	-	-
PCP	fixed: routers connected to the content source on delivery paths	threshold-based popularity	-	-	-
HVC	fixed: routers selected based on their centrality values within a network	-	-	-	-
MPC	-	threshold-based popularity	-	-	-
PbLRU	-	correlativity-based popularity	-	-	-
LCE	-	-	$p = 1$	-	-
RND	-	-	random p	-	-
PC+	dynamic: distance from the consumers and the content source on delivery paths	-	-	-	cache capacity of routers on delivery paths
HPC	dynamic: distance from the content source on delivery paths	-	-	residence time of content at routers on delivery paths	-
CAC	dynamic: routers located at the end of congested links on delivery paths	correlativity-based popularity	-	-	-

Fig. 2.13: Taxonomy of the caching policies reviewed in this thesis according to the caching criteria used.

2.3 On-path Caching: Quantitative Comparison

In this section, a quantitative comparison between a number of caching policies using the ndnSIM(v1) ns-3 simulator [Afanasyev e] is presented. A quantitative comparison of on-path caching policies is missing from the ICN literature. More specifically, a large number of caching policies proposed in the context of on-path caching, including dynamic-probability caching policies, have been evaluated under insufficient experimental information and/or unrealistic experimental parameters [Ioannou]. As an example, HPC has been evaluated using a network topology of eight routers and four servers. Such types of network topologies are not representative examples of the Internet infrastructure [Dhamdhere b, Pentikousis].

2.3.1 ndnSIM Simulator

ndnSIM is an open-source simulator written in C++, that simulates the NDN ICN architecture using the ns-3 simulation framework [Afanasyev e], a simulation platform based on discrete event scheduling. ndnSIM is one of the few actively developed and maintained ICN simulators [Tortelli], which has been updated in the following releases [Mastorakis a, Mastorakis b]. ndnSIM has been widely used by the ICN research community to evaluate a number of ICN features such as on-path caching, multi-path forwarding and request aggregation [Tortelli]. Alternative simulators, such as the ccnSim simulator [Chiocchetti] and Icarus [Saino b], are mainly focused on the performance analysis of in-network caching, without supporting all the features of the NDN architecture, such as the feature of request aggregation evaluated in chapter 4.

2.3.2 Caching Policies

In this section, a description of the adaptation of the caching policies reviewed in this thesis and adopted in this evaluation is given. The focus of this evaluation lies mainly on the dynamic-probability caching policies presented in Section 2.2.3.2. The rationale behind this choice is that dynamic-probability caching policies have been constructed to combine multiple caching criteria. Consequently, an evaluation of such caching policies will allow the researcher to collect valuable information with regard to the effect of a number of caching criteria on the performance of a network and the experience of consumers. In addition to the dynamic-probability caching policies, a fixed probability caching policy, i.e. LCE (Section 2.2.3.1), and a location-based caching policy, i.e. LCL (Section 2.2.1), are used as benchmarks. LCE is the first caching policy proposed in the context of on-path caching and the default caching policy of the NDN architecture. LCE has been widely used as benchmark in the evaluation of newer caching policies [Ioannou]. LCL is one of the first caching policies proposed in the context of on-path caching and a form of edge caching that has been widely used in the current Internet infrastructure [Fayazbakhsh, Li c]. To identify the benefit of on-path caching, an approach where no caching is applied is also adopted, called *No Caching (NC)*. The information collected will be used to guide the construction of an optimal caching policy. A complete overview of the caching policies may be found in the corresponding sections, i.e. sections 2.2.3.2, 2.2.3.1 and 2.2.1.

2.3.2.1 ProbCache+ (PC+)

PC+ [Psaras a] is an enhanced version of PC [Psaras b], proposed to overcome the unfair allocation of cache capacity between contents introduced by the y/x factor, as well as

the possibility of PC to result in a probability $p > 1$. PC+ is formulated as follows:

$$p = PC+ = ((\sum_{r=1}^{x-y+1} N_r)/(T_{tw} \times N_x)) \times (y/x)^x, \text{ where:}$$

- $\mathbf{N_r}$ is the time for which router r should cache content on delivery paths, determined by its cache size.
- $\mathbf{T_{tw}}$ is the time for which content should be cached on delivery paths.
- $\mathbf{N_x}$ is the average cache size of routers on delivery paths.
- \mathbf{x} is the number of hops traversed from a consumer to a content source.
- \mathbf{y} is the number of hops traversed from a content source to router r .

According to the authors, a router can never be aware of the cache size of every other router on the delivery paths. To solve this complication, every router assumes that each other router has a cache size equal to its own. By adopting this assumption, PC+ can now be formulated as: $((x - y + 1)/T_{tw}) \times (y/x)^x$. T_{tw} is a parameter that depends on the network topology and the network traffic and should therefore be set a priori by an ISP. Based on the evaluation conducted by the authors, small values of T_{tw} , such as $T_{tw} = 1$, result in a degradation of the caching performance, while large values of T_{tw} , such as $T_{tw} = 20$ do not seem to affect the performance of the caching policy. The default value of T_{tw} has been set equal to 10, while a T_{tw} equal to 5 has been shown to provide the best caching performance possible. By adopting this assumption, PC+ can now be formulated as: $((x - y + 1)/5) \times (y/x)^x$. Both x and y are expected to be updated at every router on the delivery paths, according to Fig. 2.9.

2.3.2.2 Hop-based Probabilistic Caching (HPC)

HPC [Wang e] is formulated as follows: $p = HPC = (1/(y + \alpha)) \times (RT_m/RT_{exp})$, if $RT_m < RT_{exp}$, and as follows: $p = HPC = 1/(y + \alpha)$, if otherwise, where:

- y is the number of hops traversed from a content source to the router that performs caching.
- α is a constant integer dependent to the network cache capacity.
- RT_m is the mean residence time of content at routers.
- RT_{exp} is the expected mean residence time of contents at routers.

Based on the evaluation conducted by the authors, the default value of α can be set equal to 1 for experimental purposes. By adopting this assumption, HPC can now be formulated as: $p = HPC = (1/(y + 1)) \times (RT_m/RT_{exp})$, if $RT_m < RT_{exp}$, and as: $p = HPC = 1/(y+1)$, if otherwise. Recalling the definition of HPC's factor $CacheWeight_{RT}$, HPC is expected to perform the best when $RT_{exp} = RT_m$, since ultimately HPC aims to predict an expected mean residence time that approaches the actual mean residence time. By adopting this assumption, HPC can now be formulated as: $p = HPC = (1/(y + 1))$.

2.3.2.3 Congestion-Aware Caching (CAC)

CAC [Badov] is formulated as follows: $p = CAC = (ctr_o / \sum_{\forall o} ctr_o) \times (S_o / \min(BW_{Up} \cup BW_{Dwn}) - S_o / \min(BW_{Dwn}))$, where:

- ctr_o is the popularity counter of object o , equal to the number of object requests received for this object.
- $\sum_{\forall o} ctr_o$ is the sum of object requests for all objects.

- \mathbf{S}_o is the size of object o .
- $\mathbf{min}(\mathbf{BW}_{\text{Up}} \cup \mathbf{BW}_{\text{Dwn}})$ is the minimum bandwidth of both the upstream links and the downstream links for the given flow.
- $\mathbf{min}(\mathbf{BW}_{\text{dwn}})$ is the minimum bandwidth of the downstream links.

Configuring different bandwidth values at the links of a network may affect the performance of the caching policies adopted in the evaluation. As an example, low bandwidth values may cause content request drops and retransmissions. Taking into account the unpredictability of dynamic-probability caching policies, such network behaviour would complicate the evaluation of the caching policies in question and the interpretation of the results. For this reason all links are configured with the same bandwidth value.

In order to adopt CAC in the evaluation, a modification has been applied where a hop-count metric is used to mimic the time saved by consumers to retrieve content, since the hop-count metric has been shown to be a good estimation of the latency metric [Kangasharju]. To mimic this behaviour, CAC is now formulated as: $p = CAC = (ctr_o / \sum_{\forall o} ctr_o) \times (y/x)$, where similar to PC, x is the number of hops traversed from a consumer to a content source and y is the number of hops traversed from a content source to the router that performs the caching decision r .

2.3.2.4 Leave Copy Everywhere (LCE)

LCE is formulated as follows: $p = 1$, meaning that content is cached at every router that it traverses. LCE does not require any modifications to be adopted.

2.3.2.5 Leave Copy Last (LCL)

LCL is neither a fixed-probability caching policy nor a dynamic-probability caching policy. LCL is a location-based caching policy that caches content at the edge routers on delivery paths. Similar to LCE, LCL does not require any modifications to be adopted.

2.3.2.6 No Caching (NC)

NC is not a caching policy. NC is used to represent a scenario where caching is disabled at the NDN routers of a network, allowing the identification of the benefit of on-path caching. Similar to LCE and LCL, NC does not require any modifications to be adopted.

2.3.3 Network Topologies

Recalling Section 1.3.1, ICN has been defined to operate on arbitrary network topologies such as scale-free topologies and Internet-like topologies derived from the Internet infrastructure. Examples of Internet-like topologies are the Rocketfuel topologies [Spring], CAIDA topologies [Caida], and the topologies generated using the GT-ITM simulator [Zegura], suggested by the ICN research community [Pentikousis]. For the purpose of this evaluation, two network topologies provided by the ndnSIM community for experimental purposes are used [Afanasyev b]. Both network topologies have been generated by the traces of the Rocketfuel topologies Tiscali AS-3257 and Exodus AS-3967, filtered to include only the routers named as “backbone” and “gateway” in the traces. Nonetheless, a modification has been applied to either network topology since the number of consumers connected to the edge routers was much smaller to the number of consumers to be used. For this purpose, all consumers have been deleted.

For ease of understanding, Fig. 2.14 and 2.15 present the Tiscali AS-3257 and Exodus

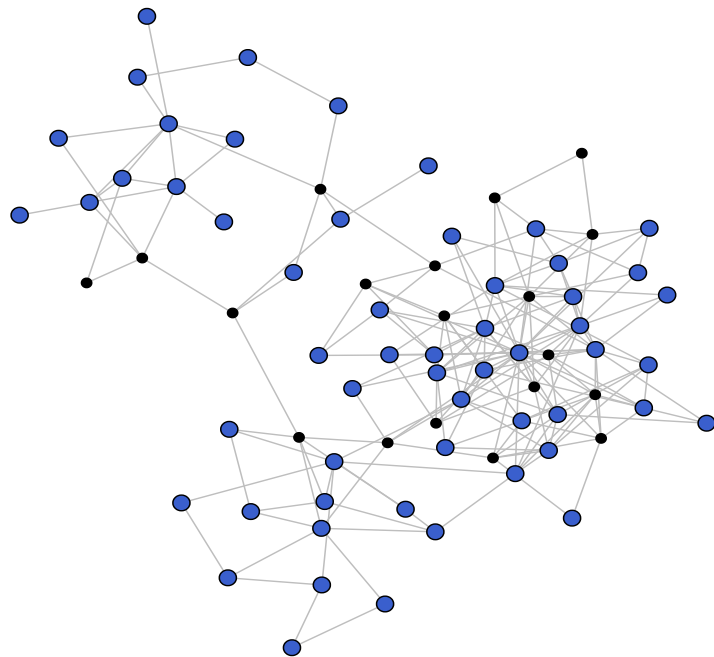


Fig. 2.14: Representation of the Tiscali Topology (AS-3257) used in the simulations, with 19 intermediate routers and 56 edge routers, indicated by the black, smaller nodes and the blue, larger nodes, respectively.

AS-3967 network topologies provided by the ndnSIM community, respectively, where the black colour is used to represent the intermediate routers and the blue colour is used to represent the edge routers. To further distinguish between the routers, the size of the edge routers has been deliberately set to be double the size of intermediate routers. In the first topology, 75 routers: 19 intermediate routers and 56 edge routers are employed. In the second topology, 97 routers: 39 intermediate routers and 58 edge routers are employed. The network topologies are similar in size, yet different in terms of the number of intermediate routers contained, with the Exodus AS-3967 network topology containing almost double the number of intermediate routers of the Tiscali AS-3257 network topology. This difference will allow the researcher to examine if the number of

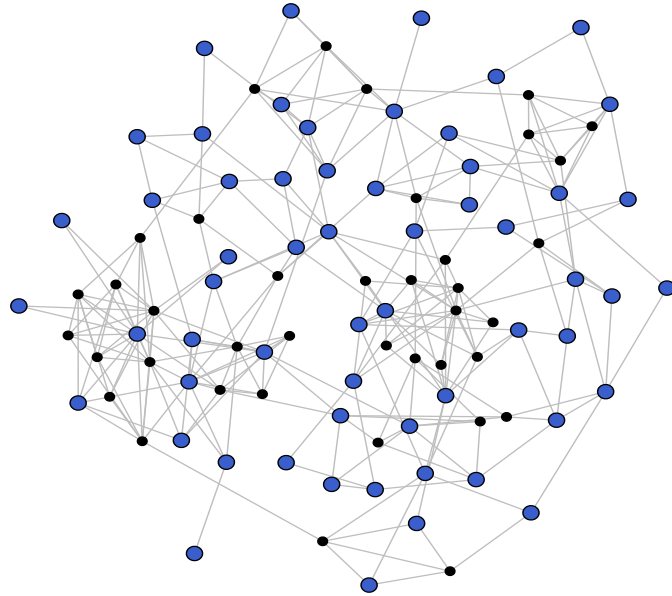


Fig. 2.15: Representation of the Exodus Topology (AS-3967) used in the simulations, with 39 intermediate routers and 58 edge routers, indicated by the black, smaller nodes and blue, larger nodes, respectively.

intermediate routers within a network may affect the performance of caching policies.

At this point it is worth noting, that equivalent sizes of network topologies have been adopted in the evaluation of a number of on-path caching policies, e.g. [Badov , Bernardini]. Following this trend, the aforementioned network topologies have been adopted to provide a quantitative comparison between a number of caching policies, yet avoid overloading the simulator (Section 2.3.4).

Each router is equipped with an NDN stack, consisting of the CS, PIT and FIB architectural components. Section 2.3.5 provides information with regard to the sizes of these components, highlighting any consequent limitations. Consumer and producer applications are installed according to the information provided in Section 2.3.4. To avoid the occurrence of content request drops and retransmissions, a link capacity of

40GB bandwidth [Psaras a] and a transmission queue of 8×10^3 content requests have been assumed within a network, based on the number of simultaneous object requests expected (Section 2.3.4). The retransmission timer has been set to be equal to 50 milliseconds. Similar to the queue size, this value has been used to ensure that both Interests and Data packets may reach any eligible entity within a network. Finally, a link delay of 10ms has been set, equal to the default value of ndnSIM [Afanasyev d].

2.3.4 Applications

With the increased popularity of multimedia applications, a number of studies have highlighted the associated rise of the Internet traffic, which is expected to exceed 3.3 zettabytes by 2021 [Gantz]. Almost 80% of this traffic is expected to be video traffic [Cisco , Sandvine], provided by large Content Service Providers (CSPs), such as YouTube. To this end, a YouTube-like traffic has been adopted in the evaluation. A YouTube-like traffic has been commonly used by the ICN research community to explore the performance of on-path caching and multi-path forwarding under realistic network topologies [Pentikousis , Rossi a]. Thus, besides the use of realistic Internet traffic, the evaluation is expected to enable comparison within the ICN research community. A summary of the traffic model used may be found in Table 2.2.

Two types of applications may be defined within an ICN architecture: a consumer/-subscriber, and a content source/producer or publisher. In this section, the definition of both applications for the purpose of this evaluation is given.

2.3.4.1 Consumer

A *consumer* application generates object requests to retrieve videos. Object requests signal the retrieval of an object resource through the expression of successive Interests.

Network Parameters	
Parameter	Symbol Value
# of routers in network	$ N $ 75, 97 [Badov ,Bernardini]
# of intermediate routers in network	$ B $ 19, 39 [Afanasyev b]
# of edge routers in network	$ G $ 56, 58 [Afanasyev b]
# of consumers at edge router i	u_i $\forall u_i, i \in G \sim U(100, 200)$ [Detti b, Li e, Saha]
# of producers in network	p 1 [Arianfar ,Psaras a]
Link capacity/bandwidth	BW 40GB [Psaras a]
Link delay	d 10ms [Afanasyev d]
Cache size in chunks/Data	cs $cse\{10^2, 10^3, 10^4\}$ [Arianfar ,Lee ,Perino]
Traffic Model Parameters	
Parameter	Symbol Value
Catalog size	$ O $ $10^6, 10^8$ [Pentikousis]
Object size in chunks/Data	C 1 [Xie ,Psaras a, Xiaoyan]
Chunk size in KB	ch 10KB [Muscariello b, Carofiglio a]
Object Popularity Parameters	
Parameter	Symbol Value
Shape of Weibull disribution	$shape$ 0.513 [Cheng ,Pentikousis]
Scale of Weibull ditribution	$scale$ 6010 [Cheng ,Pentikousis]
Request Arrival Parameters	
Parameter	Symbol Value
Mean of Exponential distribution	$mean$ 1.0 [Katsaros ,Cooper]
Upper limit of Exponential distribution	$upper$ 3.0 [Katsaros ,Cooper]

Table 2.2: Summary of the system model and the evaluation parameters used in the evaluation of the PC+, HPC, CAC, LCE and LCL caching policies.

Consumer applications can only be installed at the edge routers of a network topology. Following a uniform distribution of $mean = 150$, an average number of 150 consumers is installed at each edge router to generate realistic network traffic [Detti b, Li e, Saha]. According to the number of edge routers employed at the AS-3257 and AS-3967 network topologies, a number of 8×10^3 consumers are expected to be installed within a network.

The number of Interests expressed for an object resource should be determined by the size of the object resource divided by the chunk size. The size of an object resource is assumed to follow a normal distribution with a mean of 10MB and a standard deviation of 9.8MB [Gill , Zhou a]. The size of a chunk is defined to be equal to 10KB [Muscariello b, Carofiglio a]. Hence, an object resource is expected to be equal to 10^3 successive Interests on average. However, each successive Interest is scheduled within 0 seconds from the time the last Data was received, creating a large amount of Interests that need to be processed simultaneously by the simulator [Project b]. This introduces a significant computational and processing overhead per simulation run [Shi a]. Besides the time limitations encountered in the evaluation of full object resources, a memory consumption problem has been encountered due to the number of CS entries to be maintained per simulation run [Shi a]. Following the recommended cache size of an NDN router, i.e. equal to 10GB [Arianfar , Lee , Perino], and the chunk size of object resources, i.e. equal to 10KB [Muscariello b, Carofiglio a], each router should be equipped with a CS of the size of 10^6 chunks, where each chunk consumes 0.75KB of memory at minimum [Mastorakis b]. Similar memory problems have been reported [Shi b], while ndnSIM(v2) is expected to raise these issues further.

Due to these limitations, and since the goal of this comparison is to evaluate the effect of a number of caching criteria on the performance of a caching policy and to compare the different caching policies, the size of an object resource has been deliberately

set to be equal to a single Interest. Single-Interest object resources have been previously used in the evaluation of on-path caching policies, e.g. [Xie ,Psaras a,Xiaoyan]. This modification will allow the researcher to examine the effect of the caching criteria on a larger sample of object requests, i.e. approximately 10^4 per consumer, in a reasonable execution time. This modification will also allow the researcher to evaluate the effect of different cache sizes on the performance of a caching policy. The adaptation of single-Interest object resources is not expected to affect the performance of a caching policy since the same caching decision is expected to be applied to all chunks of an object resource. In addition to this, neither evaluation metric is dependable to the number of chunks of an object resource (Section 2.3.6). Nonetheless, full object resources are expected to highlight the differences on the performance of the caching policies in question, since the same value of an evaluation metric is expected to refer to a higher number of contents. To maintain consistency between the object size and the cache size of routers, the size of a CS has been modified to store single-Interest object resources.

A consumer decides which object resource to request from a catalog size of 10^8 object resources, based on a Weibull popularity distribution of $shape=0.513$ and $scale=6010$ [Cheng , Pentikousis], and issues object requests according to an exponential distribution of $mean = 1.0$ and $upper_{limit} = 2.0$ [Katsaros , Cooper]. The aforementioned parameters have been adopted in previous studies and suggested by the ICN research community. The aforementioned parameters will allow the researcher to mimic the behaviour of real-world consumers by introducing an average delay of 1 second, and maximum 2 seconds, between subsequent object requests, following the exponential distribution of ns-3. A catalog size of 10^8 object resources has been adopted in the evaluation of a number of on-path caching policies, following the guidelines suggested by the ICN research community to enable comparison between evaluations using *User Generated Content (UGC)* [Pen-

tikousis]. A catalog size of 10^6 object resources has also been adopted to examine the effect of the catalog size on the performance of the caching policies, assuming that the consumers within an ISP/AS network would not be interested in all UGC content.

2.3.4.2 Producer

A *producer* application generates Data with the name of the Interest received and a payload equal to 10KB [Muscariello b, Carofiglio a]. Consequently, a producer makes no distinction on whether to satisfy an Interest. Similar to consumer applications, producer applications can only be installed at the edge routers of a network topology. For the purpose of the evaluation, a single producer that possesses all object resources [Arianfar , Psaras a] is placed at an edge router with *Degree Centrality (DC)* equal to the average DC value of all edge routers within a network. The DC metric refers to the number of connections/edges of a router with the rest of the routers within a network. For both the AS-3257 and AS-3967 network topologies, the average DC value of edge routers is equal to 5. By choosing an average-connected edge router to act as a producer, the possibility of creating a bias on the performance of a caching policy with regard to the location of the producer within a network is prevented. By choosing a specific edge router to act as a producer, the collection of statistics over the same set of routers between the individual simulation runs is guaranteed.

2.3.5 Architectural Components

In this section, a description of the details to be defined with regard to the operation of the CS, PIT and FIB architectural components of the NDN architecture is given.

- **CS:** Referring to Section 1.3.1, the size of a CS has been suggested to be equal to

10GB. Nonetheless, this suggestion has been based on the retrieval of full object resources and the expression of successive Interests. Since the size of an object resource has been reduced from 10^3 Interests to a single Interest, the size of a CS needs to be reduced accordingly to maintain consistency and to avoid affecting the performance of caching policies. For this reason, the size of a CS has been set to be equal to 10MB. Since each chunk is equal to 10KB, the size of a CS is equal to 10^3 chunks/Data. A CS size of 10^2 and 10^4 chunks have also been considered to examine the effect of the CS size on the performance of caching policies.

- **PIT:** Similar to a CS, the size of a PIT needs to be bounded to avoid raising scalability and deployability concerns. However, such size restrictions may affect the performance of a network and trigger the occurrence of Interest drops and retransmissions [Abu , Virgilio], which would in turn complicate the evaluation of the caching policies in question and the interpretation of the results. For these reasons, and since the scalability and deployability concerns of a PIT are beyond the scope of this thesis, no size restrictions are defined for the operation of a PIT.
- **FIB:** A FIB does not apply to any numerical parameters or modifications to its fundamental design that need to be defined for the purpose of the evaluation. Nonetheless, a FIB has been assumed to propagate Interests according to the *Best Route (BR)* forwarding strategy, a forwarding strategy that utilises FIB entries registered by the SPR protocol. This assumption ensures that a single path will be used for the retrieval of an object resource during the evaluation. The rationale behind this choice is that multi-path forwarding has been shown to increase both the network traffic and the cache replacements, while lessening the benefits of on-path caching [Rossi a, Rossini a]. Since the goal of this evaluation is the exploration

of the effect of the caching criteria used on the performance of a caching policy, rather than the effect of the forwarding mechanism on the performance of a caching policy, the consideration of multi-path forwarding has been omitted.

2.3.6 Evaluation Metrics

In this section, a description of the evaluation metrics used in the evaluation of the caching policies is given. The evaluation aims to provide valuable information with regard to the effect of a number of caching criteria on the performance of a network and the experience of consumers. To this end, and due to the integration of on-path caching with the forwarding mechanism of ICN, both *cache-based metrics* and *network-based metrics* are used to quantify the benefit of a caching policy to reduce both the intra and inter-network traffic of an ISP/AS network, the main objectives of this thesis.

2.3.6.1 Cache-based Metrics

Cache-based metrics are used to measure the effectiveness of a caching policy by measuring whether it is able to cache and maintain the desired content. Hence, cache-based metrics measure the ability of a caching policy to satisfy successive Interests locally at an NDN router and reduce the network traffic generated due to the propagation of Interests and Data. Cache-based metrics are calculated at all routers within a network.

- **Cache-Hits Ratio (CHR):** The *cache-hits ratio* metric refers to the number of Interests satisfied by a router, normalised by the number of Interests received at this router, whether these have been satisfied or not. This metric indicates the offload of a producer by a caching policy. Preferable caching policies should result in an increase of this metric. The cache-hit ratio metric is also referred to as the

cache-hits rate or the *cache-hits probability*.

- **Caching Frequency (CF):** The *caching frequency* metric refers to the number of Data cached at a router, normalised by the number of Data received at this router. This metric shows the tendency of a caching policy to cache new content. The caching frequency metric is not a representative metric to the performance of a caching policy by its own. Nonetheless it provides an indication of the memory consumption at routers. Preferable caching policies should result in a reduction of this metric and to an increase of the cache-hit ratio metric and/or to a reduction of the hop-count metric described in the following section.
- **Cache-Evictions Ratio (CER):** The *cache-evictions ratio* metric refers to the number of cache replacements that occur in the cache of a router using a replacement policy, normalised by the number of Data cached at this router. This metric indicates the exploitation of the existing cached content. Preferable caching policies should result in a reduction of this metric. This metric is also referred to as the *cache-replacements rate* or the *cache-evictions rate*.

2.3.6.2 Network-based Metrics

Network-based metrics are used to quantify the performance of a caching policy with regard to the network topology. For the purpose of this evaluation, the *hop-count* metric is used to mimic the delay experienced by consumers, since the hop-count metric is a good estimation of the latency metric [Kangasharju]. The hop-count metric refers to the number of routers traversed by an Interest and/or a Data packet on the forwarding and/or the delivery paths. Hence, the hop-count metric indicates the amount of traffic generated within a network caused due to the propagation of Interests and

Data [Kangasharju ,Li a]. Caching policies should result in a reduction of this metric. Network-based metrics are calculated at the consumer applications within a network.

2.3.7 Evaluation Results

In this section, the evaluation results of PC+, HPC, CAC, LCE and LCL are presented. The evaluation results of NC are also presented for comparison purposes. For ease of reference, the term *approach* is used to refer to the caching policies and evaluation system. For ease of comparison, the mean and standard deviation (SD) values of the evaluation metrics of an approach are used, with the SD values being presented on top of the mean values of an evaluation metric. The SD values are used as a complementary descriptive measurement to the mean values to identify the variance on the behaviour of caching policies and the effect of the evaluation parameters on their performance [Hassani]. This form of representation of the mean and SD values may exceed the maximum value of an evaluation metric. Similar to former studies in the context of on-path caching [Bernardini ,Janaszka], both values have been derived from the evaluation results of 10 simulation runs with regard to the parameters presented in Table 2.2.

Fig. 2.16 and 2.17 illustrate the mean hop-count and SD values of an approach with regard to the catalog size $|O|$ and the cache size cs for the AS-3257 network topology and the AS-3967 network topology, respectively, where $|O| = \{10^6, 10^8\}$ and $cs \in \{10^2, 10^3, 10^4\}$. The mean hop-count of NC is equal to 4 for the AS-3257 network topology and equal to 5 for the AS-3967 network topology. The mean hop-count of CAC is equal to 3 and 4, respectively. The mean hop-count of PC+ is equal to that of CAC when $cs \in \{10^2, 10^3\}$, yet 2 hops lower when $cs = 10^4$ for the AS-3257 network topology. The mean hop-counts of HPC, LCE and LCL are equal for all cache sizes. The mean hop-count of HPC is equal to that of CAC when $cs = 10^3$, yet 1 hop higher

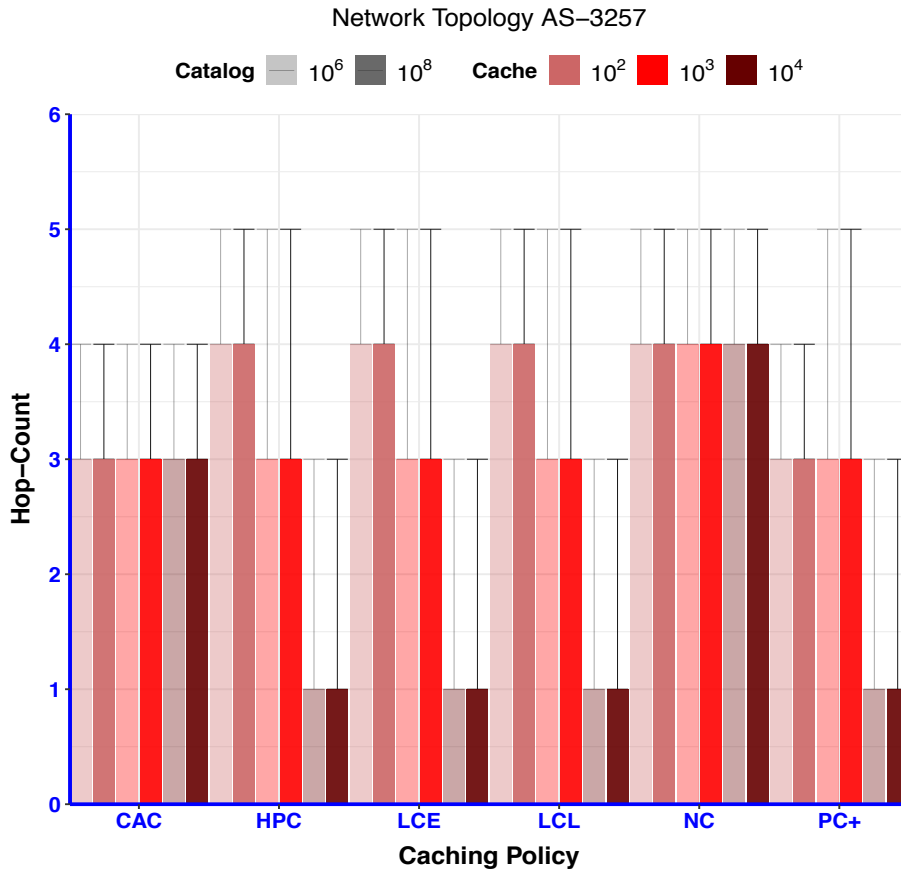


Fig. 2.16: Representation of the mean hop-count and SD values of an approach for the AS-3257 network topology, when $|O| = \{10^6, 10^8\}$ and $cs \in \{10^2, 10^3, 10^4\}$.

and 2 hops lower when $cs = 10^2$ and $cs = 10^4$, respectively, for the AS-3257 network topology. The mean hop-count of PC+ is equal to the mean hop-count of LCL for all cache sizes and equivalent to that of HPC for the AS-3967 network topology. The mean hop-count of LCL is equal to 4, 2 and 1 hops when $cs \in \{10^2, 10^3, 10^4\}$, respectively. The mean hop-count of HPC is 1 hop higher to that of LCL when $cs = 10^2$. In contrast to the alternatives, the mean hop-count of LCE does not decrease as the cache size increases. The SD values of all caching policies are between 1-2 hop-counts for either

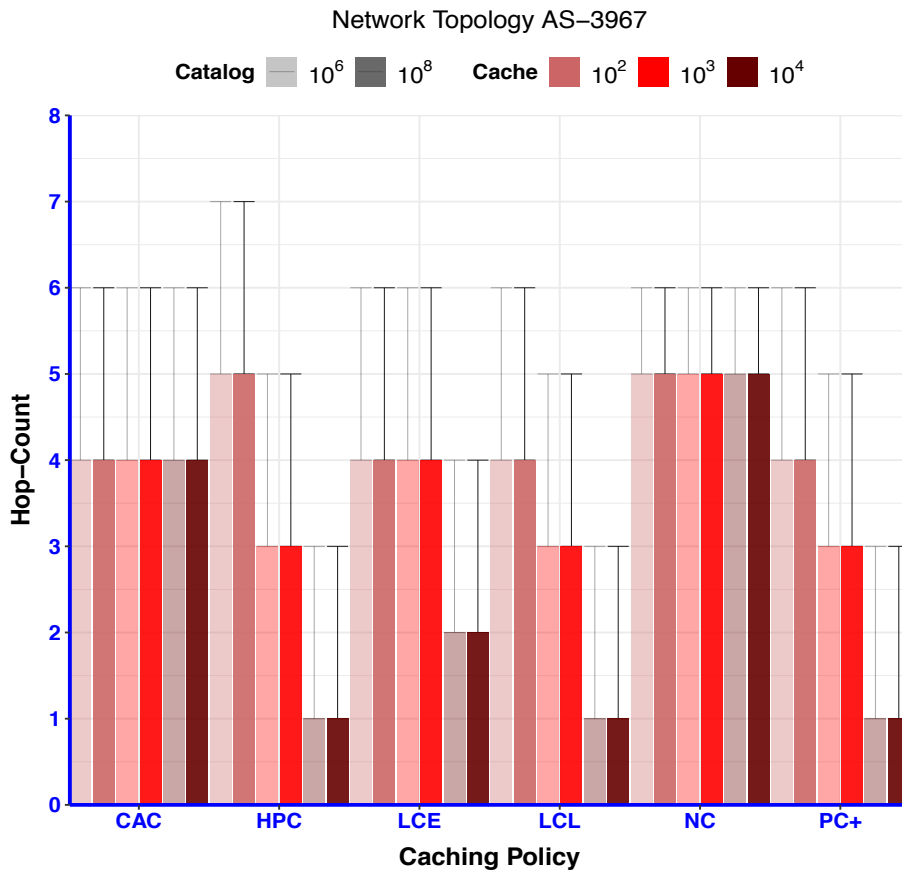


Fig. 2.17: Representation of the mean hop-count and SD values of an approach for the AS-3967 network topology, when $|O| = \{10^6, 10^8\}$ and $cse\{10^2, 10^3, 10^4\}$.

network topology. Besides NC and CAC, whose SD values remain constant as the cache size increases, an increase of 100% of the SD values is shown for the remaining caching policies as the cache size increases from $cs = 10^2$ to $cs = 10^3$ for the AS-3257 network topology. Moreover, the SD values of all caching policies indicate a higher variance of the metric for the AS-3967 network topology, compared to the AS-3257 network topology.

According to both figures, a number of conclusions may be withdrawn as follows:

- The hop-count metric of a caching policy, e.g. HPC and LCL, decreases as the

cache size increases, since more contents are cached at a router, allowing content requests to be satisfied within a shorter distance compared to the distance to a content source. This behaviour increases the variance of the hop-count metric.

- An increase in the cache size of routers does not guarantee a lower hop-count metric, since a caching policy decides whether to cache content according to its caching criteria. This claim may be verified by consulting the mean hop-count of CAC and its behaviour, according to which content is cached only if it is popular, and regardless of the cache size. A caching policy similar to CAC is PC+.
- The hop-count metric of a caching policy decreases as the number of routers on the forwarding and the delivery paths increases, since the number of routers to which caching may be applied and the overall cache capacity increase as well. This claim may be verified by consulting the mean hop-count of either caching policy, except CAC, which is 1 hop lower to that of NC for the AS-3967 network topology, compared to the AS-3257 network topology, indicating a higher variance of the metric. For HPC, LCL and PC+ this difference is visible when $cs = 10^3$. For $cs = 10^4$ this difference is not visible since a large amount of cache capacity is already available. For LCE this difference is visible when $cs = 10^2$.
- The hop-count metric of LCE is affected from both an increase in the number of routers to cache content and the overall cache capacity, resulting into faster content distribution [Bernardini ,Chai ,Cho]. In contrast to its benefit when $cs = 10^2$, a decrease in its hop-count metric is concluded for the AS-3967 network topology, compared to the AS-3257 network topology, when $cs \in \{10^3, 10^4\}$. This behaviour may be explained based on the fact that LCE will cache more redundant content.

- The hop-count metric of a caching policy is independent of the catalog size, since the difference between the catalog sizes considered is insignificant to affect this metric. This claim may be verified by the fact that a right skewed distribution has been used by consumers to issue object requests, limiting the number of object resources that are expected to affect the performance of a caching policy. An additional factor that may have limited the impact of this parameter is the high amount of object requests generated in the simulations, which is expected to extinguish any negligible differences. A final factor that may have limited the impact of this parameter is the adaptation of single-Interest object resources in the evaluation of the caching policies in question. Nonetheless, consulting chapter 4 and the evaluation of the request aggregation mechanism proposed in this thesis, no impact has been encountered on its performance and besides the fact that full object resources have been adopted in the evaluation.

Fig. 2.18 and Fig. 2.19 illustrate the mean CHR and SD values of an approach with regard to the catalog size $|O|$ and the cache size cs for the AS-3257 network topology and the AS-3967 network topology, respectively, where $|O| = \{10^6, 10^8\}$ and $cs \in \{10^2, 10^3, 10^4\}$. The results are equivalent for both network topologies. The mean CHR of CAC is equal to 6% for the AS-3257 network topology, and equal to 4% and 5% for the AS-3967 network topology, when $cs = 10^2$ and $cs \in \{10^3, 10^4\}$, respectively. These values are 2-3% higher than the mean CHR of the alternatives when $cs = 10^2$, for either network topology. In contrast to CAC, the mean CHR of the remaining caching policies increases as the cache size increases, with LCL performing 1-2% higher to LCE when $cs \in \{10^3, 10^4\}$, and HPC performing the same as PC+. The mean CHR of LCL when $cs \in \{10^3, 10^4\}$ is equal to 14% and 49% for the AS-3257 network topology, and equal to

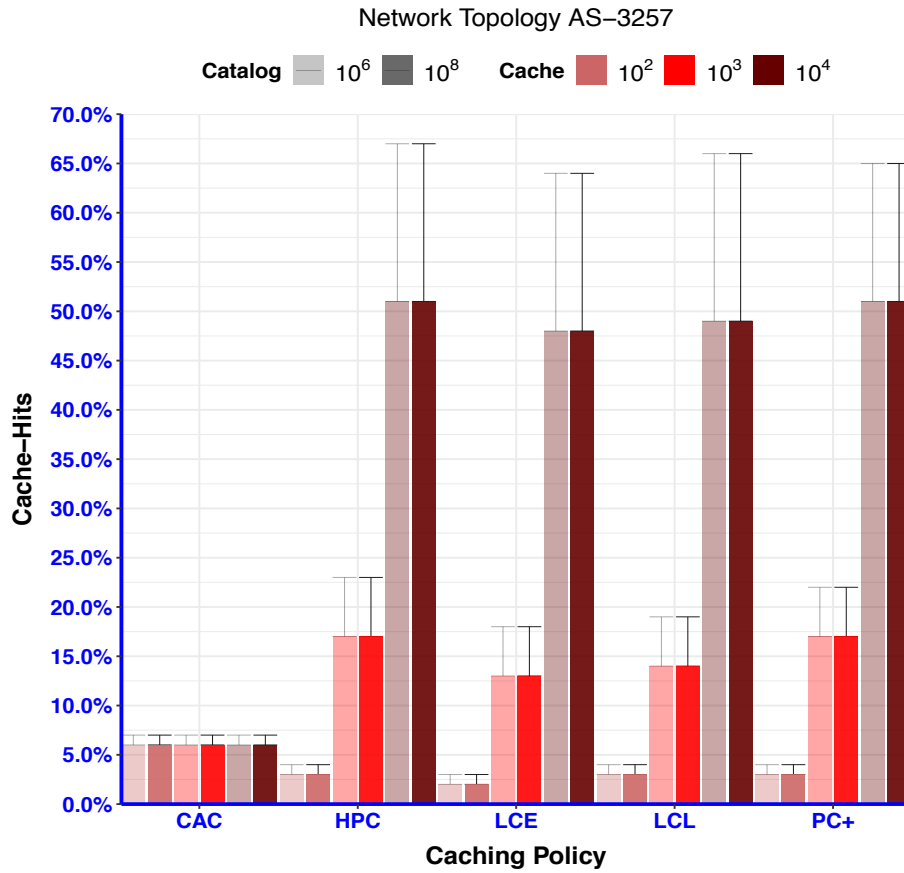


Fig. 2.18: Representation of the mean CHR and SD values of an approach for the AS-3257 network topology, when $|O| = \{10^6, 10^8\}$ and $cs \in \{10^2, 10^3, 10^4\}$.

13% and 44% for the AS-3967 network topology, respectively. The mean CHR of HPC is equal to 17% and 51%, and equal to 16% and 48% for the same cache sizes and network topologies. Besides CAC, whose SD values remain constant as the cache size increases, the SD values of the alternatives are equivalent and between 1-20%, while a higher SD value is shown as the cache size increases for either network topology. Moreover, the SD values of the caching policies indicate a higher variance of the metric for the AS-3967 network topology, compared to the AS-3257 network topology.

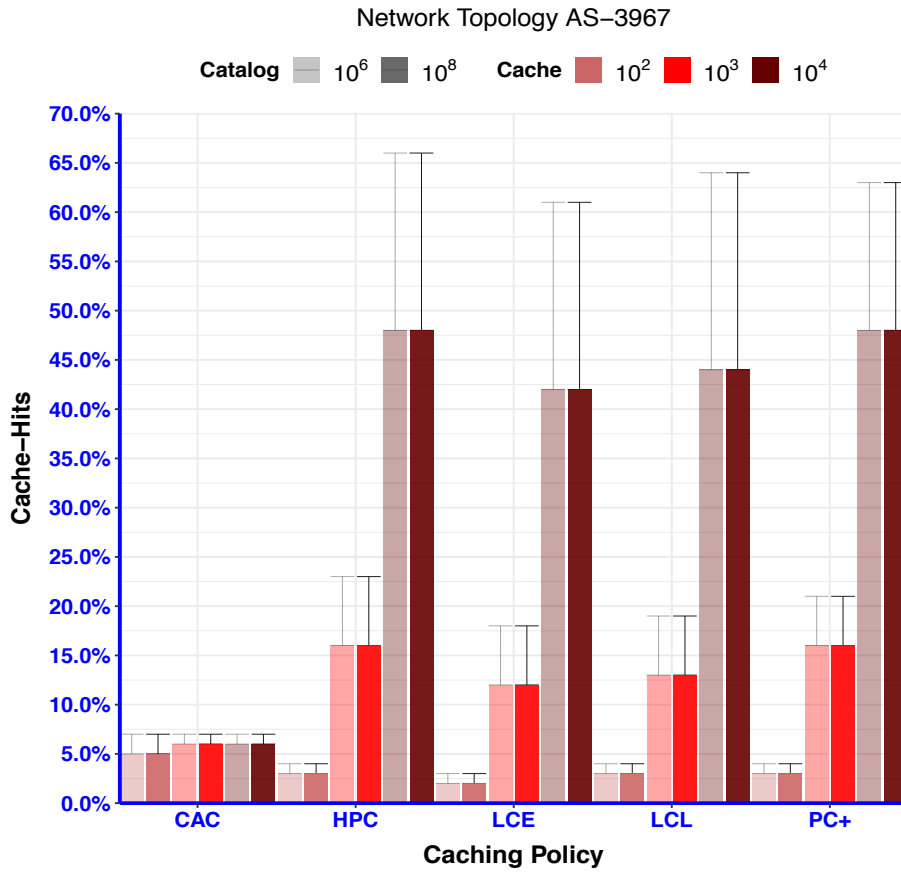


Fig. 2.19: Representation of the mean CHR and SD values of an approach for the AS-3967 network topology, when $|O| = \{10^6, 10^8\}$ and $csc\{10^2, 10^3, 10^4\}$.

According to both figures, a number of conclusions may be withdrawn as follows:

- Similar to the impact of the cache size and the catalog size on the hop-count metric of a caching policy, an equivalent impact is observed for the CHR metric.
- The CHR metric of a caching policy decreases, while its SD value increases, as the number of routers on the forwarding and the delivery paths increases. This claim may be verified as follows: a decrease of 1-6% and an increase of 1-3% of the mean CHR and SD values of all caching policies is shown for the AS-3967 network

topology, compared to the AS-3257 network topology. This difference is higher for higher cache sizes, and when $cs \in \{10^3, 10^4\}$.

- An increase in the number of routers on the forwarding and the delivery paths increases both the number of routers to cache content and the overall cache capacity for LCE, since LCE makes no distinction on what content to cache in a router. This increase benefits the CHR metric of LCE when the cache size of routers is small, i.e. $cs = 10^2$, yet higher cache sizes increase content redundancy, causing a reduction of this metric [Che , Laoutaris , Wong].
- An increase in the number of routers on the forwarding and the delivery paths increases the probability of caching content closer to consumers for caching policies whose caching decision is based on the number of hops traversed by the content requests and/or the content replies, such as HPC and PC+. Yet, it also decreases the caching probability on delivery paths, causing a reduction of the CHR metric. This claim may be verified by comparing Fig. 2.20 and 2.21, which illustrate the mean CF and SD values of an approach with regard to the evaluation parameters considered. Consulting the results, the mean CF of HPC and PC+ is 1-4% lower for the AS-3967 network topology, compared to the AS-3257 network topology.

Besides the mean CF and SD values of HPC and PC+ presented in Fig. 2.20 and 2.21, the mean CF and SD values of CAC, LCE and LCL are also presented:

- The mean CF of LCE is equal to 100%, since LCE caches content at every router.
- The mean CF of CAC is equal to 0%, which may be explained by the fact that the mean CF of CAC is considerably low, i.e. of 10^{-4} magnitude, and the fact that all

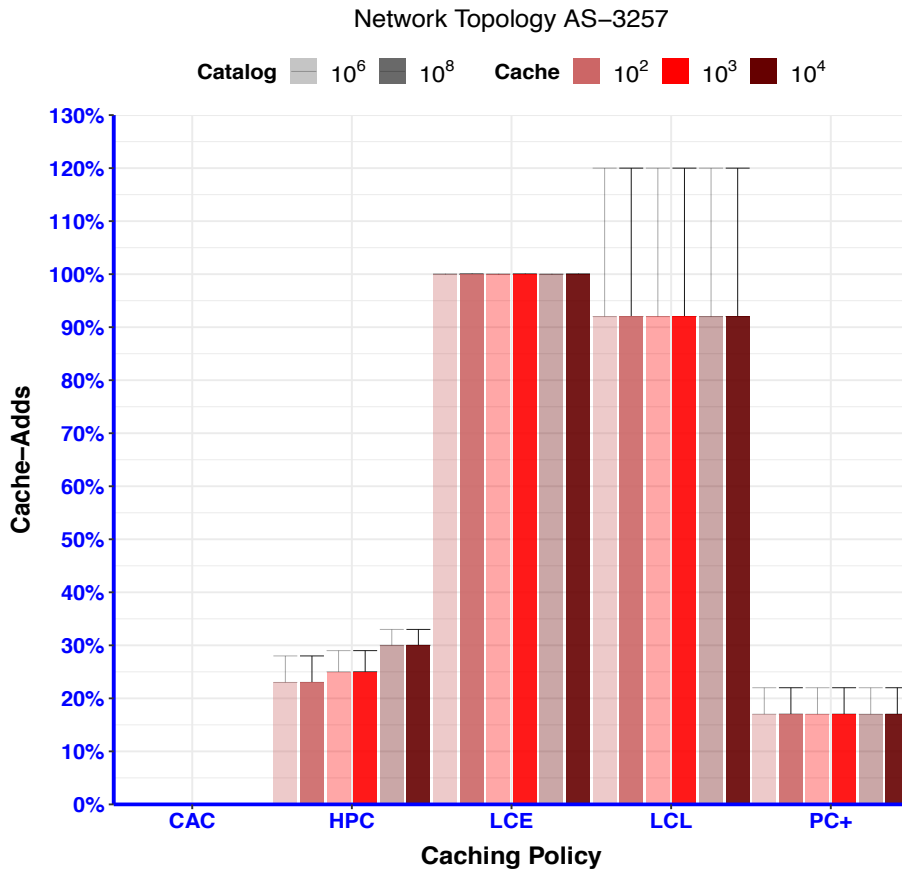


Fig. 2.20: Representation of the mean CF and SD values of an approach for the AS-3257 network topology, when $|O| = \{10^6, 10^8\}$ and $cs \in \{10^2, 10^3, 10^4\}$.

cache-based metrics have been rounded to two decimal places. This claim may be verified by consulting the other evaluation metrics of CAC, such as CER.

- The SD values of both LCE and CAC are equal to 0%, since the mean CF values of both caching policies remain constant.
- The mean CF of LCL is equal to 92% for the AS-3257 network topology and equal to 86% for the AS-3967 network topology. This difference may be explained as follows: To compare the performance of the caching policies, all mean values have

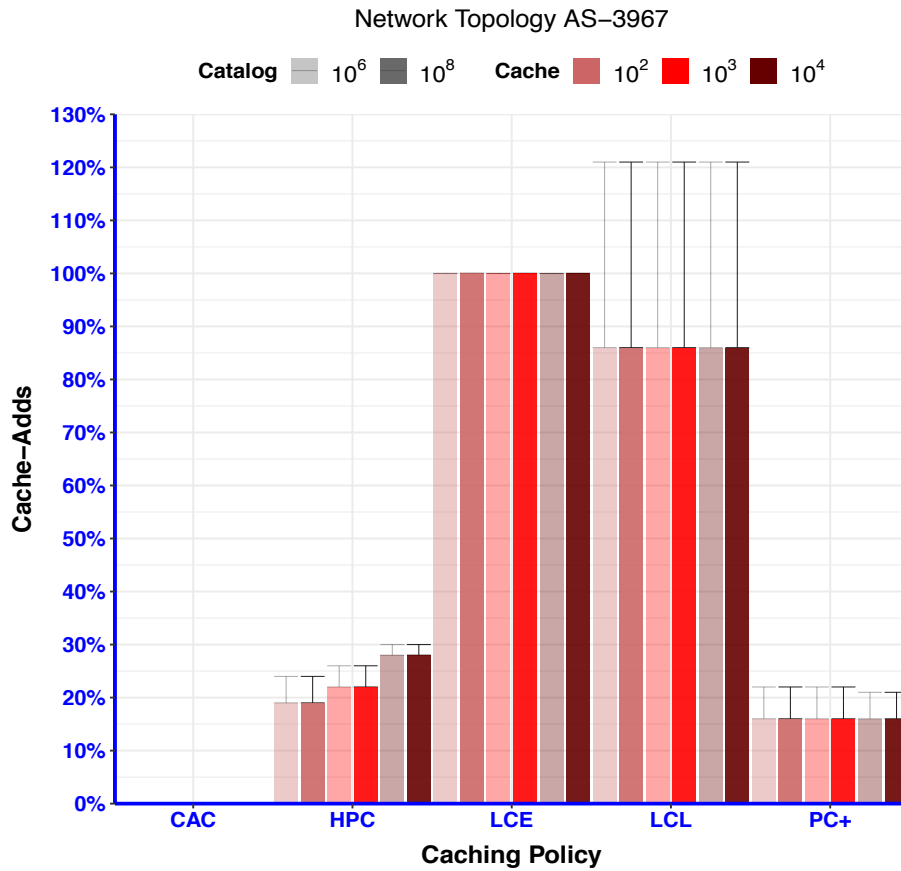


Fig. 2.21: Representation of the mean CF and SD values of an approach for the AS-3967 network topology, when $|O| = \{10^6, 10^8\}$ and $c\epsilon \in \{10^2, 10^3, 10^4\}$.

been calculated based on the values of the routers in the sets of shortest paths used in simulations, calculated using the SPR protocol and the BR forwarding strategy. This number is equal to 61 routers for the AS-3257 network topology and equal to 67 routers for the AS-3967 network topology. Since this number is lower for the AS-3257 network topology, a higher CF metric is concluded. A mean $CF < 100\%$ value indicates that not all routers on the sets of shortest paths have cached content, indicating a higher SD value for shorter CF values.

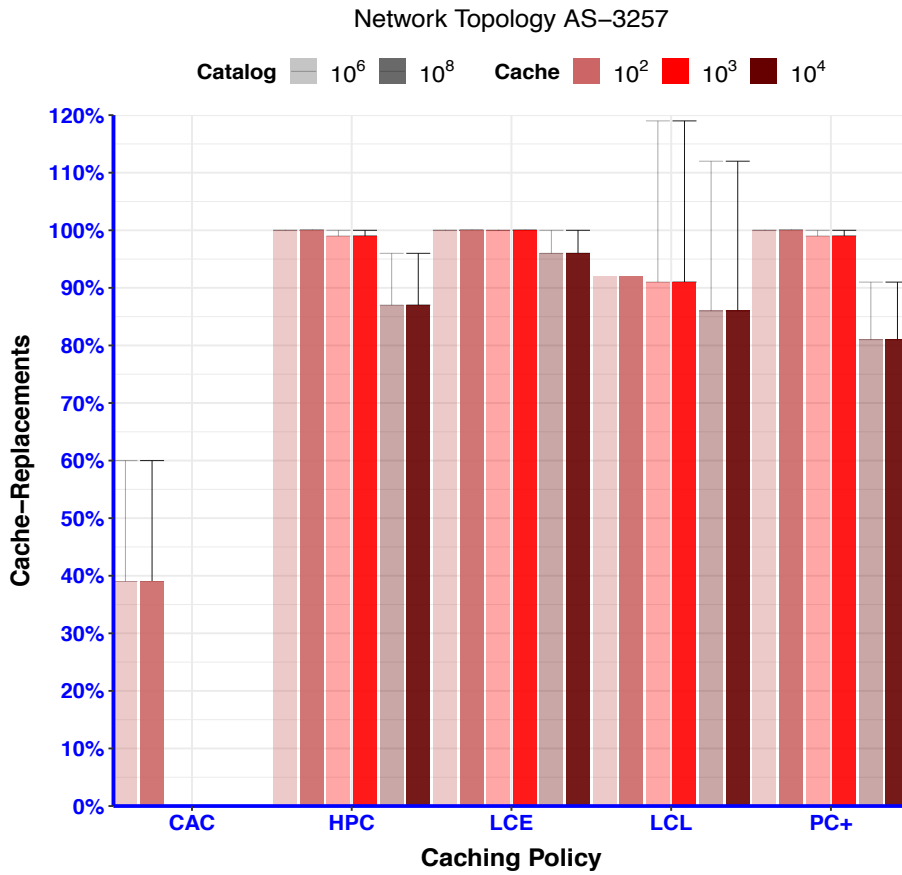


Fig. 2.22: Representation of the mean CER and SD values of an approach for the AS-3257 network topology, when $|O| = \{10^6, 10^8\}$ and $cs \in \{10^2, 10^3, 10^4\}$.

- The SD values of HPC and PC+ range between 2-6%.

Fig. 2.22 and 2.23 illustrate the mean CER and SD values of an approach with regard to the catalog size $|O|$ and the cache size cs for the AS-3257 network topology and the AS-3967 network topology, respectively, where $|O| = \{10^6, 10^8\}$ and $cs \in \{10^2, 10^3, 10^4\}$. Referring to both figures, a similar pattern to the CF metric is shown:

- The mean CER of LCE is equal to 100% when $cs \in \{10^2, 10^3\}$, and equal to 96% when $cs = 10^4$. This difference may be explained by the fact that a higher cache

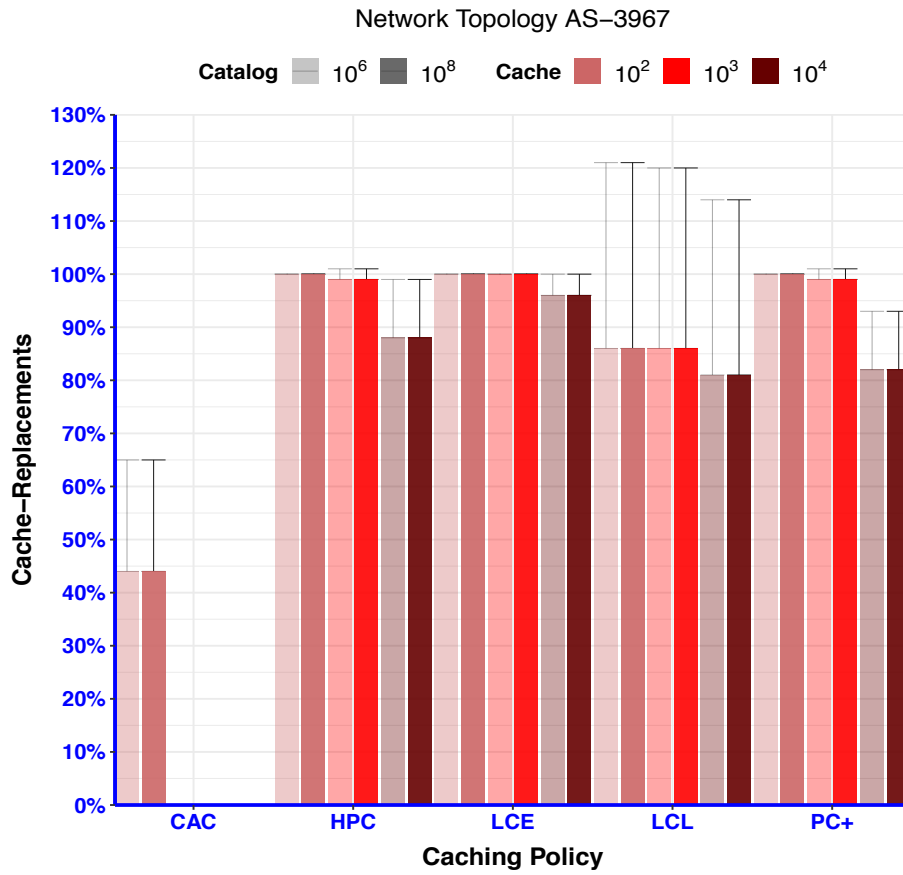


Fig. 2.23: Representation of the mean CER and SD values of an approach for the AS-3967 network topology, when $|O| = \{10^6, 10^8\}$ and $cs \in \{10^2, 10^3, 10^4\}$.

size allows more contents to be cached at routers without the necessity to replace content, increasing the variance of this metric from 0% to 4%.

- The mean CER and SD values of CAC are equal to 0% when $cs \in \{10^3, 10^4\}$. This outcome may be explained by the fact that the CF values of CAC are considerably low and the fact that all cache-based metrics have been rounded to two decimal places. Even though such low CF values may cause cache replacements when $cs = 10^2$, i.e. equal to 39% for the AS-3257 network topology and equal to 44% for

the AS-3967 network topology, higher cache sizes are not expected to affect this metric. The SD values for the equivalent CER values of CAC are equal to 21%.

- The mean CER and SD values of HPC and PC+ are equal to 81-100% and 0-11%, respectively, while a decrease of 12-19% and an increase of up to 11% of these values is shown as the cache size increases. Moreover, a maximum increase of 1% is shown between the values concluded for the AS-3257 network topology, compared to the AS-3967 network topology. The mean CER of HPC is 6% higher compared to the mean CER of PC+ when $cs = 10^4$, and equal if otherwise. This difference may be explained by consulting the mean CF values of HPC, which is 3-13% higher compared to the mean CF values of PC+.
- In contrast to the remaining caching policies, whose CER values are mainly independent of the network topology, the mean CER of LCL is equal to 86-92% for the AS-3257 network topology and equal to 81-86% for the AS-3967 network topology, respectively, while the equivalent SD values are up to 28% and 35%. Similar to the CF metric, this outcome may be explained based on the number of routers on the sets of shortest paths used in simulations for either network topology.

2.3.8 Lessons Learned

In this section, a summary on the performance of a caching policy following the results presented in Section 2.3.7 is given for PC+, HPC, CAC, LCE and LCL. This summary will help the researcher to construct the dynamic-probability caching policy proposed in chapter 3. Hence, the aim of this section is to highlight the effect of the criteria used by a caching policy to perform a caching decision on the performance of a network and the experience of consumers, as follows:

- LCE is a fixed-probability caching policy that caches content at every router on delivery paths, making no distinction on the content or the location of routers. Due to caching redundant content, LCE may faster exploit an increase in the number of routers on the delivery paths and/or the network cache capacity, providing faster content dissemination compared to the alternatives. Yet, an increase of the same parameters may degrade its performance, if the rate of redundant content is high. Consequently, LCE is sensitive to the changes of the parameters considered.

In addition to the lower performance of LCE for the hop-count metric, i.e. up to 1 hop, this difference is noticeable for all cache-based metrics. Besides CAC, LCE results in the lowest CHR metric and the highest CER metric compared to the alternatives, i.e. up to 6% and 15%, respectively, while utilising up to 100% of the network cache capacity. Hence, LCE performs the worst of all caching policies.

- LCL is a location-based caching policy that caches content at a fixed location, i.e. the edge routers on delivery paths, to reduce the number of hops to be traversed by both content requests and content replies within a network. LCL benefits by an increase in the number of routers on delivery paths. LCL performs better than LCE for the hop-count metric, and the CHR and CER metrics, i.e. up to 1 hop, and up to 2% and 15%, respectively. The results highlight the importance of caching content closer to consumers and the fact that a higher benefit may be achieved even though caching is applied to less routers within a network.
- PC+ and HPC are dynamic-probability caching policies whose functionality is based on the number of hops traversed by the content requests and/or the content replies. Both caching policies yield a higher performance gain than the alternatives for the CHR and the CF metrics. More specifically, an increase of up to 6% is shown

for the CHR metric by caching 70-84% less content at the routers within a network. The performance of the caching policies with regard to the hop-count metric varies, with PC+ performing the best among the alternatives and HPC performing 1 hop higher or lower than the alternatives, depending on the evaluation parameters.

- PC+ performs better than HPC. This claim may be verified by consulting the results of their hop-count metric when $cs = 10^2$, which is 1 hop higher for HPC, regardless of the network topology. Moreover, the same CHR is concluded for both caching policies, besides the fact that PC+ caches 3-13% less content than HPC at the routers within a network. Hence, PC+ performs the best of all caching policies. The exact reason that led to the performance difference between PC+ and HPC is unclear, since PC+ differs from HPC in a number of ways. Nonetheless, a conclusion that may be safely made based on the definition of the caching policies adopted for the evaluation conducted in this chapter, is that the caching probability of PC+ is considerably lower to that of HPC. This difference indicates that HPC is more likely to cache redundant content on the delivery paths compared to PC+.
- Even though both HPC and LCL cache content according to the location of a router on delivery paths, HPC results in a higher performance gain than LCL. Besides the lower performance of HPC than LCL for the hop-count metric and the AS-3967 network topology, when $cs = 10^2$, i.e. equal to 1 hop, HPC results in a higher performance gain for both the CHR and CF metrics. More specifically, HPC results in a higher CHR of 2-4% compared to LCL, while caching 58-69% less content at the routers within a network. The results highlight the benefit of utilising the same caching criterion into the caching decision process dynamically.
- CAC is a dynamic-probability caching policy whose functionality, based on its

definition adopted for the evaluation conducted in this chapter, is based on the number of hops traversed by the content requests and the content replies on the forwarding and delivery paths, and the criterion of content popularity. CAC yields a higher performance gain than the alternatives when the cache size of routers is small, i.e. $cs = 10^2$. This claim may be verified by consulting the results of the metrics of hop-count, CHR and CF. CAC results in an equal performance gain, or 1 hop higher than the alternatives for the hop-count metric. Similar, CAC yields an increase of 3-4% of the CHR metric against the alternatives by caching a negligible amount of content at the routers within a network. The results highlight the importance of the factor of content popularity into the caching decision process.

- CAC is unable to exploit the network cache capacity when higher cache sizes are utilised, i.e. $cs \in \{10^3, 10^4\}$. This drawback derives from the fact that the caching decision of CAC is bounded by the factor of content popularity. This claim may be verified by consulting the results for both the hop-count metric and the CHR metric, which have shown to remain constant, while both the CF and CER metrics are negligible, i.e. equal to 0%.

The conclusions presented in this section complement the observations drawn from the taxonomy of the caching policies reviewed in this thesis, presented in Section 2.2.4.

2.3.9 Discussion

In this section, a discussion on the parameters used in the evaluation of the PC+, HPC, CAC, LCE and LCL caching policies is given. The discussion aims to highlight the limitations of the system model adopted in Section 2.3.7 and its impact on the results.

Consulting Section 2.3, a number of evaluation parameters have been selected to

avoid the occurrence of content request drops and retransmissions, and to provide an optimal, homogeneous system model. Such parameters include the link capacity and the transmission queue. Even though the rationale behind this choice is commonly acceptable for evaluation purposes, the performance of either caching policy may be affected by modifying these parameters. The amount of impact is dependent to a number of factors, including the network topology and forwarding paths, the number of consumers and the content popularity distribution. As an example, edge routers may be connected to lower capacity links within an ISP/AS network than backbone routers. Such network topologies may either degrade or benefit the performance of caching policies that aim to cache content at the edge routers within a network to reduce the content delivery times of consumers, depending on the location of the producer that hosts the content [Badov]. More specifically, a lower delay is expected to be introduced to the content delivery times of consumers due to the propagation of both content requests and content replies over the links where producers are connected to backbone routers, compared to the edge routers. Regardless of the location of a producer within a network and its impact on the experience of consumers, caching content closer to consumers is expected to reduce both the intra and the inter-network traffic of an ISP/AS network.

Another evaluation parameter that may affect the performance of a caching policy and the caching policies reviewed in this thesis is the network topology. Consulting Section 2.3.4.1, and the limitations encountered in the evaluation of full object resources using the ndnSIM simulator, the selection of this parameter has been limited to the AS-3257 network topology and the AS-3967 network topology provided by the ndnSIM community for experimentation. Both network topologies are highly connected since their average DC value is equal to 5. Smaller network topologies are expected to benefit the performance of on-path caching policies, besides the caching criteria used. This

claim may be verified by consulting both the hop-count metric and the CHR metric of either caching policy for the AS-3257 network topology, compared to the AS-3967 network topology, which is slightly bigger. Highly connected network topologies are also expected to benefit the performance of on-path caching since a higher number of content requests is expected to be propagated through the same set of routers on the forwarding and the delivery paths. Hence, although such network topologies may provide some intuition on the behaviour of caching policies and the caching criteria used, a variety of network topologies, including large network topologies [Chai ,Wang 14] and disconnected network topologies [Badov] should also be considered to generalise the results.

Besides the network topology parameter, the limitations encountered in the evaluation of full object resources have also limited the size of an object resource adopted to a single Interest. Even though according to Section 2.3.4.1, the adaptation of single-Interest object resources is not expected to affect the performance of a caching policy, the adaptation of full object resources is expected to highlight the differences on the performance of the caching policies in question. This claim may be explained based on the fact that the values of the evaluation metrics presented in Section 2.3 are expected to be applied to a higher number of contents, i.e. to each chunk of an object resource.

An additional evaluation parameter expected to affect the performance of a caching policy is the popularity distribution adopted by consumers to issue object requests. Content popularity distributions define which content should be cached within an ISP/AS network [Carofiglio a,Rossi a]. In this thesis, a Weibull popularity distribution of $shape=0.513$ and $scale=6010$ has been adopted, according to the parameters suggested by the ICN research community for experimentation with UGC traffic [Pentikousis]. Adopting this distribution has allowed the researcher to mimic a YouTube-like network traffic. Since YouTube is one of the most popular CSPs, and since almost 80% of the

network traffic is expected to be video traffic by 2021 [Gantz], a YouTube-like traffic is expected to provide valuable information with regard to the benefits of a caching policy in real deployments. Nonetheless, additional parameters and popularity distributions may be used to research the behaviour of on-path caching, and the impact of the caching criteria used on the performance of the network and the experience of consumers.

Another evaluation parameter expected to affect the performance of a caching policy is the catalog size. Even though Section 2.3.7 concludes that the difference encountered between the catalog sizes adopted in the evaluation has been negligible to affect the performance of the caching policies in question, a higher catalog size is expected to lessen the benefits of either caching policy [Dehghan]. A higher catalog size is also expected to highlight the differences between the caching policies and the ability to cache and maintain popular content. These claims may be explained based on the fact that a higher number of contents is expected to be competing for the same cache entry.

2.4 Request Aggregation: The Mechanism

Request aggregation is a mechanism of the NDN architecture that aims to limit the propagation of Interests to a content source(s) to a single Interest per content. Request aggregation has been characterised as a native flow-control mechanism and a security mechanism of NDN that aims to reduce the network traffic and prevent the expansion of DoS attacks to both consumers and content sources. In contrast to the publish-subscribe paradigm and a naming scheme, request aggregation is not a fundamental component of an ICN architecture. Following the naming granularity of NDN, request aggregation is applied on a chunk naming granularity, while its operation may be described as follows: Upon the arrival of an Interest and if a PIT entry that matches the name of the Interest

exists, the PIT entry will be updated according to the information included in the Interest, such as its lifetime and nonce. The face at which the Interest arrived will also be added to the interfaces list of the PIT entry and the Interest will be discarded.

2.4.1 Efficiency Analysis

In this section, a description of the analysis to identify the efficiency of a request aggregation mechanism within an NDN network and the conclusions drawn from this analysis is given. These conclusions will help the researcher to construct the mechanism proposed in chapter 4. By enhancing the efficiency of the request aggregation mechanism, an increase in the performance of the mechanism proposed in chapter 4 is also expected.

Even though the motivation behind the benefits of the request aggregation mechanism have been explicitly defined [Jacobson , Yi b, Zhang a], these statements lack experimental and/or analytical evidence. To this end, two analytical models have been employed to analyse the probability of request aggregation occurring within an NDN network and its efficiency [Dabirmoghaddam , Ghali], described as follows:

- In the first analytical model the probability of request aggregation occurring at the edge routers is modelled. According to the authors, the rationale behind this choice is that the benefit of a request aggregation mechanism is expected to be higher at the edge routers, compared to the rest of the routers within a network. This assumption comes in contrast to the results obtained from the simulations held using the second analytical model, where the benefit of the mechanism of request aggregation has been shown to be higher for the routers placed closer to content sources, rather than the consumers. This controversial outcome may be explained according to the fact that cache misses that occur at the edge routers,

or at the routers placed closer to consumers, will be progressively aggregated at the routers placed closer to content sources.

- In the second analytical model the probability of request aggregation occurring at all routers within an NDN network is modelled. For this purpose an iterative algorithm that expands the design of a model similar to the first analytical model to construct a hierarchical network is used. Each iteration corresponds to a single level of the hierarchical network, while the results of each lower level are fed into the next level in the hierarchy. Consumers are placed at the bottom level of the network hierarchy, while a content source is placed at the top level. The design differences between the two analytical models are likely to be the reason why the efficiency of request aggregation has been shown to be lower for the first analytical model compared to the second one. As an example, the probability of request aggregation occurring within an NDN network is estimated between 0.05% and 0.15% for the first analytical model and between 0.5% and 5% for the second.

Despite of the design differences between the two analytical models, the following two factors have been concluded to be critical to the performance of request aggregation:

- **Entry Lifetime:** The time interval for which an entry remains alive in the PIT. This is due to the fact that request aggregation depends on the likelihood that a successive Interest for the same content is about to occur, while a matching entry is still alive in the PIT. Since this time interval is expected to be of the order of tens to hundreds of milliseconds [Ghali , Carofiglio a], PIT entries result in a relatively small probability of request aggregation occurring within a network.
- **On-path Caching:** The ability of an NDN router to cache content on delivery

paths. This is due to the fact that request aggregation depends on stateful forwarding. Since on-path caching allows Interests to be satisfied locally at a router and prevents them from being propagated upstream, the functionality of a request aggregation mechanism is cancelled. Consequently, the probability of request aggregation occurring within an NDN network when on-path caching is applied refers neither to the popular nor to the unpopular contents within a network, but to the contents that lie in between, i.e. the not-so-popular contents. This behaviour may be explained as follows: Since popular contents have a lower probability to be cached, it is more likely to cancel the benefit of request aggregation. On the contrary, unpopular contents have a lower probability to be cached and to be requested within time intervals of the order of milliseconds.

2.4.2 Discussion

The efficiency of a request aggregation mechanism is negatively affected by the efficiency of a caching policy, which may be affected by a number of parameters such as the cache size of routers within a network. The higher the cache size is, hence the efficiency of a caching policy, the lower is the efficiency of request aggregation. Additionally to the effect of on-path caching on the efficiency of request aggregation, the efficiency of request aggregation is mainly affected by the small-time intervals for which request aggregation may be applied within an NDN network. This time interval is equal to the time between the arrival of an Interest and the creation of a PIT entry, and the arrival of Data and the deletion of the corresponding entry. Based on the currently existing technologies, this time interval is expected to be of the order of tens to hundreds of milliseconds [Ghali ,Carofiglio a], resulting in a relatively small probability of occurrence.

Considering the aforementioned complications, and the catalog sizes on which ICN

architectures are challenged to operate, i.e. 10^{12} , the probability of request aggregation occurring and providing a noticeable difference on the performance of an ICN architecture is questionable.

2.5 Summary

In this chapter, a thorough description of the NDN architecture has been given. Moreover, a thorough description of the feature of on-path caching and the mechanism of request aggregation in NDN have been presented. Approaches to on-path caching have been categorised according to the caching criteria used. Based on these criteria, and a number of additional criteria, a taxonomy that highlights the trends in on-path caching has been constructed. Complementary, a number of on-path caching policies have been evaluated based on simulations. Consulting the results, a discussion that highlights the benefits of the caching criteria used by a caching policy on the performance of the network and the experience of consumers has been employed. Following the description on the feature of on-path caching, a description on the efficiency of a request aggregation mechanism has been provided. Last, a discussion on the probability of request aggregation occurring within an NDN network has been held.

Chapter 3

PLbC: A Popularity and Location-based Caching Policy for On-path Caching

In this chapter, a caching policy that aims to reduce the traffic within a local ISP/AS network, and beyond the boundaries of this network to the Internet infrastructure is proposed, called *Popularity and Location-based Caching (PLbC)* policy. By reducing the intra and inter-network traffic and by satisfying the Interests of consumers locally, PLbC is expected to reduce the content delivery times experienced by consumers, thus satisfying both the objectives of this thesis.

PLbC is a dynamic-probability caching policy that aims to address the problem of on-path caching by selectively caching content that is expected to satisfy a higher number of Interests closer to consumers. This way, PLbC reduces the number of hops to be traversed within a network and the network traffic generated due to the propagation

of Interests and Data. For this purpose, PLbC utilises both the criterion of content popularity and the criterion of the location of routers on delivery paths, by incorporating them into the caching decision process to construct a probability. To determine which content to cache, an additional architectural component has been adopted, called *Statistical Information Table (SIT)*. SIT aims to collect statistical information of the number of requests issued for an object resource and the maximum number of requests. To determine at which router to cache content, both the Interest and Data packets have been modified to include information about the number of hops that each of them has traversed on the forwarding path and the deliver path(s), respectively.

PLbC is a lightweight caching policy based on the utilisation of local information at an NDN router. PLbC is assumed to be performed at every caching-enabled NDN router within an ISP/AS network by utilising a utility function that determines the benefit of caching the content locally. For this purpose, PLbC is composed of the *Popularity* (P_f) and the *Location* (L_f) factors, formulated as follows: $p = PLbC = P_f \times L_f$. The P_f factor refers to the comparison of the popularity counter of the object resource to which the content belongs against the popularity counter of the most popular object resource, based on the information collected in the SIT table. The L_f factor refers to the distance to be traversed by successive Interests in order to retrieve the same content from its original content source, thus the benefit of caching the content locally in hop-counts.

The design of PLbC has been based on the information collected and the conclusions drawn by both the taxonomy of the caching policies reviewed in this thesis, and the evaluation of a number of caching policies presented in chapter 2.

3.1 Context and Motivation

Recalling Section 2.2.4 and the taxonomy of caching policies, both the criterion of the location of routers and the criterion of content popularity have been broadly used within the context of on-path caching. Referring to Section 2.3.8 and the quantitative comparison of a number of caching policies based on simulations, both caching criteria have been proven to benefit the performance of a caching policy. The conclusions verify the importance of the aforementioned caching criteria indicated by the observations drawn from the taxonomy, summarised as follows:

- Caching policies whose functionality is based on the criterion of the location of routers on delivery paths, incorporated dynamically into the caching decision process, are expected to yield a higher performance gain compared to caching policies that utilise the same caching criterion in a static, fixed way.
- Caching policies whose functionality is based on the criterion of content popularity, incorporated into the caching decision process by utilising correlativity-based popularity calculations, are expected to yield a higher performance gain compared to caching policies that do not utilise this caching criterion.

At this point it is worth noting, that neither the effect of path-scale or threshold-based popularity calculations on the performance of a caching policy have been evaluated against the benefit of correlativity-based popularity calculations. This choice may be justified as follows: According to the existing literature, path-scale popularity calculations fail to address the cache pollution problem effectively, while threshold-based popularity calculations may frequently result into outdated calculations and unutilised cache capacity due to the challenging definition of a threshold [Bernardini , Janaszka , Li b]. Hence,

the focus of the evaluation has been caching policies that utilise correlativity-based popularity calculations, such as CAC. Although similar to CAC, PbLRU is a caching policy that utilises correlativity-based popularity calculations, PbLRU requires the construction of a decreasingly ordered list according to the values of the popularity counters of contents. For this purpose, a sorting algorithm is executed upon the arrival of a content request, which is expected to introduce a significant computational overhead at routers. Due to this reason, PbLRU has been excluded from the set of simulations presented in chapter 2, the conclusions of which have motivated the design of PLbC.

According to the aforementioned conclusions it is clear that neither the criterion of the position of a router on deliver paths, nor the criterion of content popularity should be omitted from the caching decision process. Moreover, incorporating these caching criteria into the caching decision process in a dynamic way instead of a static, fixed way is preferable. To this direction, the aim is to utilise these caching criteria by incorporating them into the caching decision process of PLbC in a similar way to the one adopted by the PC+, HPC and CAC dynamic-probability caching policies.

To avoid the complications encountered due to the design adopted by either of these caching policies with regard to the caching criteria in question, a number of modifications have been applied. Examples of such complications are the storage of redundant content on delivery paths performed by HPC and the underutilisation of the cache capacity of a network performed by CAC, caused due to the low values concluded by the correlativity-based popularity calculations of the content popularity factor. More into this, the aim is to adopt a number of design guidelines that have been shown to benefit the performance of a caching policy. An example of such guidelines is to cache content closer to consumers.

3.2 Design and Implementation

In this section, a description of the design of PLbC, including the modifications applied to the original NDN architecture as described in Section 2.1, is presented. More specifically, the modifications applied to the CS, PIT and FIB architectural components are presented, while the functionality of the SIT architectural component is introduced. Besides the architectural components, the modifications applied to the Interest and Data packet formats are presented. A description of the upstream and the downstream processes followed at an NDN router that supports the PLbC caching policy is also given.

3.2.1 Statistical Information Table (SIT)

To support the functionality of PLbC, an additional architectural component has been adopted that collects statistical information with regard to the number of requests issued for an object resource, called *Statistical Information Table (SIT)*. Similar to CS, SIT is assumed to be an NDN architectural component of a caching-enabled router, rather than a mandatory component of the NDN architecture.

The structure of an SIT table is relatively simple. Each entry is constructed by bounding the name of an object resource, also referred to as a *prefix*, to the corresponding popularity counter ctr_o , indicated by the number of object requests received for this object resource at an NDN router. The reception of an object request is determined upon the reception of an Interest and if an Interest refers to the first chunk of an object resource. Besides the popularity counter of an object resource, the maximum number of requests received at a router for any object resource $max_{\forall o \in O} ctr_o$ is also maintained. Nonetheless, this entry is no different to the rest of the entries in an SIT table.

An SIT table collects statistical information about object resources for which the

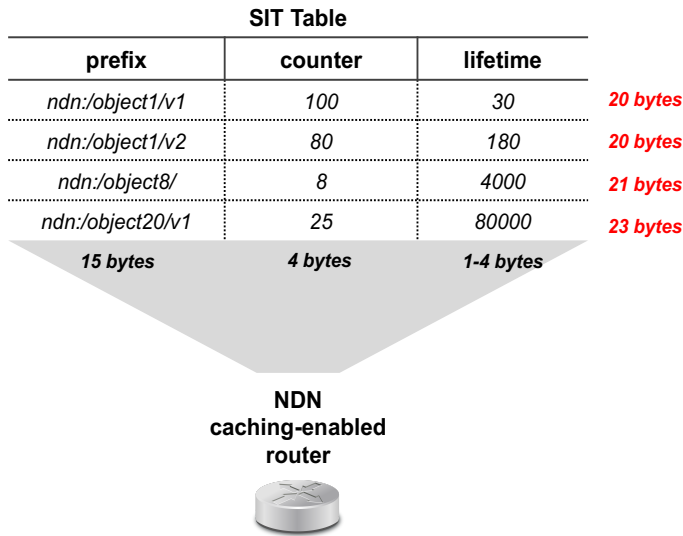


Fig. 3.1: Abstract representation of the functionality of PLbC at an NDN caching-enabled router, illustrating the structure and memory consumption of an SIT entry.

requesting activity remains active within the local ISP/AS network, allowing this way old information to expire. For this purpose, a time counter is applied to an SIT entry, similar to the lifetime field of a PIT entry used to discard old Interests that have not been satisfied yet (Section 2.1.4). This design decision is also expected to maintain the size of an SIT table to a minimum. If otherwise, the size of SIT is expected to grow linearly with the number of individual objects requested within an ISP/AS network. The optimisation of an SIT table has been defined as future work in Section 5.3.1.

The definition of a time counter is similar to the definition of a time interval Δt for which the popularity counters of the individual object resources and/or contents of an object resource are expected to be populated. The definition of a time interval able to capture the dynamics of a network is dependent to a number of factors, including the network topology, the number of consumers and the content popularity distribution. The collection and the analysis involved into this large set of information result into

a complicated system that may only be maintained by an NMM, an ISP, or a similar entity [Ahmed ,Domingues ,Sourlas a]. For this reason, a common technique used in the context of on-path caching is to define an infinite Δt [Badov ,Cho ,Xie]. This approach has been adopted in the evaluation of PLbC, where infinite time counters are used. The use of infinite time counters allows the SIT table to collect sufficient information with regard to the network traffic patterns of a local ISP/AS network and the evaluation of the performance of PLbC when the content popularity counters have converged.

For ease of understanding, Fig. 3.1 provides an abstract representation of the functionality of PLbC at a caching-enabled NDN router, illustrating the structure and the memory consumption of an SIT entry in bytes, following the definitions below:

- **prefix:** A content identifier used to identify an object resource, for which an Interest that refers to the first chunk of an object resource has been received. A prefix is constructed no different to the name field of an Interest or a Data packet presented in Section 2.1.2. Thus, the size of a prefix can not be bounded and it is dependent to the size of the individual name components contained. Nonetheless, and in contrast to the definition of a name content identifier that follows a chunk naming granularity, a prefix should not include the sequence number (seq) of a chunk of an object resource. If that number exists, it should be discarded.

Following the definition of the name content identifier $ndn:/object1/v1/s0$ in Fig. 2.1, the equivalent prefix is: $ndn:/object1/v1$. The size of this prefix is equal to 15 bytes using a UTF-8 encoding scheme. For ease of simplicity, similar name prefixes have been used to present the structure and memory consumption of SIT table in Fig. 3.1. The size of an SIT entry is presented in bytes besides each entry.

- **counter:** A counter is a non-negative integer within the range $\{0, 2^{32} - 1\}$, used

to indicate the number of object requests received for an object resource at an NDN router. Based on the number of requests for UGC content [Cha ,Figueiredo ,Yu a], it is assumed that a non-negative value of 4-bytes should be sufficient to represent the content popularity of an object resource within a local network.

- *lifetime*: A time counter that once expired, triggers the removal of the entry. A lifetime is initialised and updated upon the arrival of an object request. A lifetime is related to the network traffic patterns of the local ISP/AS network determined by the network topology, the number of consumers, the content popularity distribution and the distribution adopted by consumers to issue object requests. Due to these dependencies, a lifetime should be defined a priori by an ISP. Hence, similar to a prefix, the size of this field can not be bounded. Nonetheless, a lifetime of a few hours may be suggested, e.g. between half an hour and 2 hours, to capture the popularity of contents. This suggestion has been based on the time intervals used by former studies to predict the popularity of UGC content and the fact that the time required for a content to reach its popularity peak may vary from a few hours to days [Ahmed]. Yet, a good estimation of the content's popularity may be obtained from the number of requests for a content in the first few days of its lifecycle [Cha ,Figueiredo].

3.2.2 Utility Function and Caching Decision

PLbC is executed at every caching-enabled router within an ISP/AS network utilising the statistical information collected in the SIT table and the information collected within the Interest and Data packets with regard to the number of hops that each of them has traversed on the forwarding path and the deliver path(s), respectively. This is achieved

Algorithm 1 Calculate PLbC utility function

Require: NDN caching-enabled router r

- 1: Read $ctr_o, max_{\forall o \in O} ctr_o$ from *SIT* at r
 - 2: $P_f \leftarrow ctr_o / max_{\forall o \in O} ctr_o$
 - 3: Read hp_u, hp_d from *Data* at r
 - 4: $P_l \leftarrow hp_d / hp_u$
 - 5: $p \leftarrow P_f \times L_f$
 - 6: $rnd \leftarrow rnd(0, 1)$
 - 7: **if** $rnd \leq p$ **then**
 - 8: Cache *Data* in *CS* at r
 - 9: **else**
 - 10: Discard *Data* at r
 - 11: **end if**
-

by utilising a utility function that determines the benefit of caching the content locally at an NDN router r . For this purpose, PLbC is composed of the *Popularity* (P_f) and the *Location* (L_f) factors, formulated as follows: $p = PLbC = P_f \times L_f$. For ease of understanding, Algorithm 1 provides the pseudocode of PLbC described as follows:

- **P_f** : The P_f factor is calculated for every content received at every caching-enabled NDN router on a delivery path, where the term content refers to a chunk of an object resource o that traverses through router r . The P_f factor refers to the comparison of the popularity counter of the object resource to which the content belongs ctr_o , against the maximum popularity counter for any object resource $max_{\forall o \in O} ctr_o$, based on the information collected in SIT. For this purpose, P_f is defined as: $P_f = ctr_o / max_{\forall o \in O} ctr_o$, where $P_f \in [0, 1]$. The definition of the P_f

factor has been based on the experimental evaluation of a former definition given by the researcher. According to this definition, the popularity counter of the object resource to which the content belongs ctr_o was compared against the summation of the popularity counters of all object resources in the SIT table to construct a probability, i.e. $ctr_o / \sum \forall o \in O ctr_o$. Similar to the popularity factor of CAC, this definition of the P_f factor has been shown to result in an underutilisation of the cache capacity of routers. Hence, an alternative approach has been adopted to enforce comparison between contents and to increase the caching probability.

The rationale behind the object naming granularity adopted in the design of the popularity counters ctr_c of PLbC is twofold: First, the size of SIT would be significantly higher if a popularity counter for each chunk of an object resource was to be used, increasing both the SIT lookup time and the memory consumption at a router. Consulting the object size and the chunk size of the YouTube-like traffic adopted in the evaluation of a number of caching policies in chapter 2, an additional number of 10^3 SIT entries should be maintained per object resource. Second, the use of chunk-based popularity counters is pointless when SPR is used, since all chunk-based popularity counters of an object resource will be equal.

- **L_f** : Similar to the P_f factor, the L_f factor is calculated for every content received at every caching-enabled NDN router on a delivery path. The aim of this factor is to reduce the number of hops to be traversed by successive Interests issued to retrieve the same content, by caching the content closer to consumers. By doing so, the network traffic caused due to the propagation of Interest and Data packets within the local ISP/AS network is expected to be reduced.

The definition of the L_f factor has been based on the *hops-up* (hp_u) and the *hops-*

down (hp_d) metrics included in the Interest and Data packets formats, respectively, as follows: $L_f = hp_d/hp_u$, where $L_f \in [0, 1]$. The hp_u metric refers to the number of hops traversed by an Interest on the forwarding path, from the consumer that issued the Interest to an NDN router that possesses a replica of the content in its CS or a content source. The hp_d metric refers to the number of hops traversed by the corresponding Data, from the time the packet was issued and until it was received by the router that performs the caching decision r . Consequently, the L_f factor indicates the benefit of caching content at the local router r instead of retrieving the content from a content source, in hop-counts.

The L_f factor is equivalent to the CacheWeight factor of PC. Even though, this factor has been modified in PC+ to ensure that the caching probability p is proportional to the distance of a router from a content source and that PC+ $\in [0, 1]$, PLbC does not share the same design problem. In addition to this, and since the design of PLbC is still likely to result in low values of the content popularity factor, the CacheWeight factor has been adopted intact to ensure that the overall caching probability remains higher. Even though, the $CacheWeight_y$ factor of HPC could be used to guarantee this prerequisite, the evaluation results for this caching policy suggest that HPC is likely to increase the rate of redundant content cached on delivery paths. To avoid this complication, the option of adopting the $CacheWeight_y$ factor of HPC into the caching decision of PLbC has been rejected.

Upon the reception of Data at a caching-enabled NDN router and after the utility function for the corresponding content has been calculated, an NDN router will perform a caching decision according to the probability p .

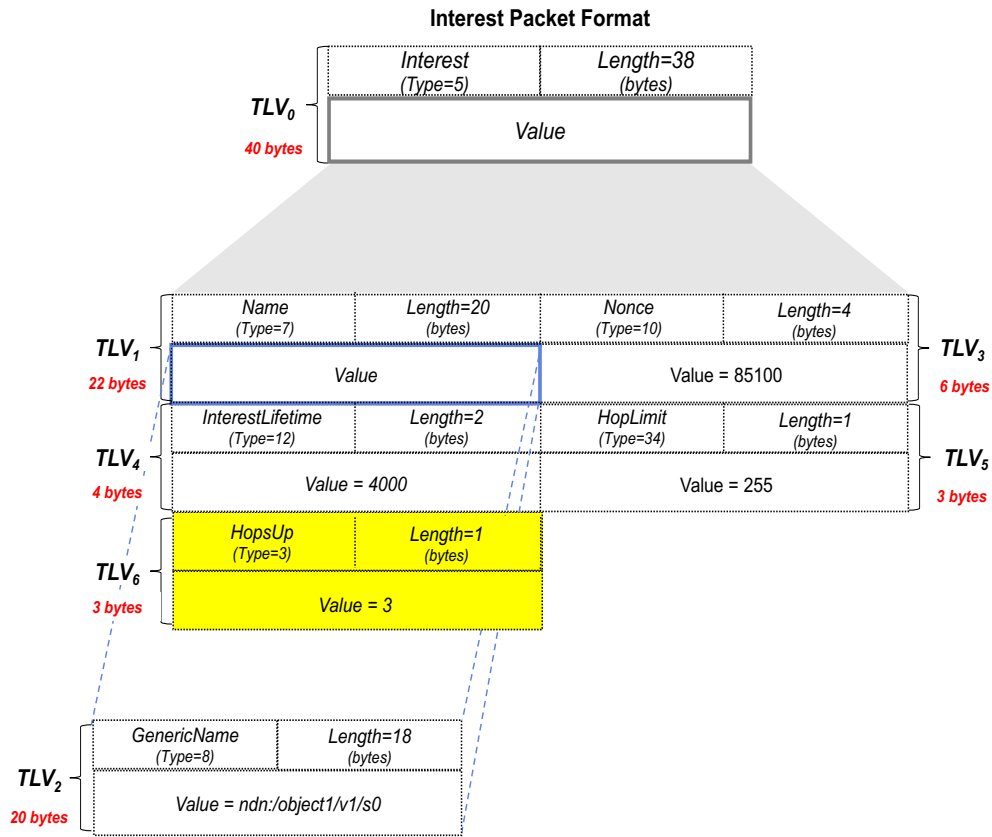


Fig. 3.2: Representation of the fields of an Interest packet necessary and/or related to the context of this thesis for the operation of PLbC at an NDN caching-enabled router.

3.2.3 Interest and Data Packets

To determine at which router to cache content, PLbC has based its operation on the collection of information with regard to the number of hops that each Interest-Data packet pair has traversed on the forwarding path and the deliver path(s), respectively. For this purpose, both the Interest and Data packets have been modified to include this information. This is a common approach in the context of on-path caching, where the Interest and/or Data packets may be modified to collect information related to a caching

decision (Section 2.2.4). Besides the inclusion of this information, both packet formats are equivalent to the packet formats introduced in Section 2.1.2. For this reason, the focus of this section lies on highlighting the differences between the Interest and Data packet formats. For ease of understanding, Fig. 3.2 and 3.3 provide a representation of the packet formats assuming that both packets have traversed 3 hops.

- **Interest** : To support the collection of information with regard to the number of hops traversed by an Interest, an additional field called *hopsup*, as described in Section 3.2.2, has been injected into the Interest packet format. This is similar to the *hoplimit* field of an Interest packet used to indicate the number of routers that an Interest may traverse before being discarded. Similar to the *hoplimit*, and since the value of *hopsup* is limited by the maximum value supported by the *hoplimit* field, *hopsup* is defined to be a non-negative integer within the range $\{0, 2^8 - 1\}$. Hence, the size of this field is equal to 1 byte.
- **Data** : The information collected by an Interest will be lost once the Interest gets satisfied by either an NDN router or the original content source. Hence, to maintain its value, as well as the number of hops traversed by the corresponding Data, two additional fields, called *hopsup* and *hopsdown* have been injected into the Data packet format. Upon the satisfaction of an Interest, the value of its *hopsup* field will be copied into the equivalent field of the Data packet.

NDN follows a symmetric forwarding strategy where both Interest and Data packets are propagated through the same sequence of routers and paths. Consequently, both the *hopsup* and the *hopsdown* fields of a Data packet are defined to be equivalent to the *hopsup* field of an Interest packet, i.e. both fields are defined to be non-negative integers within the range $\{0, 2^8 - 1\}$, equal to 1 byte.

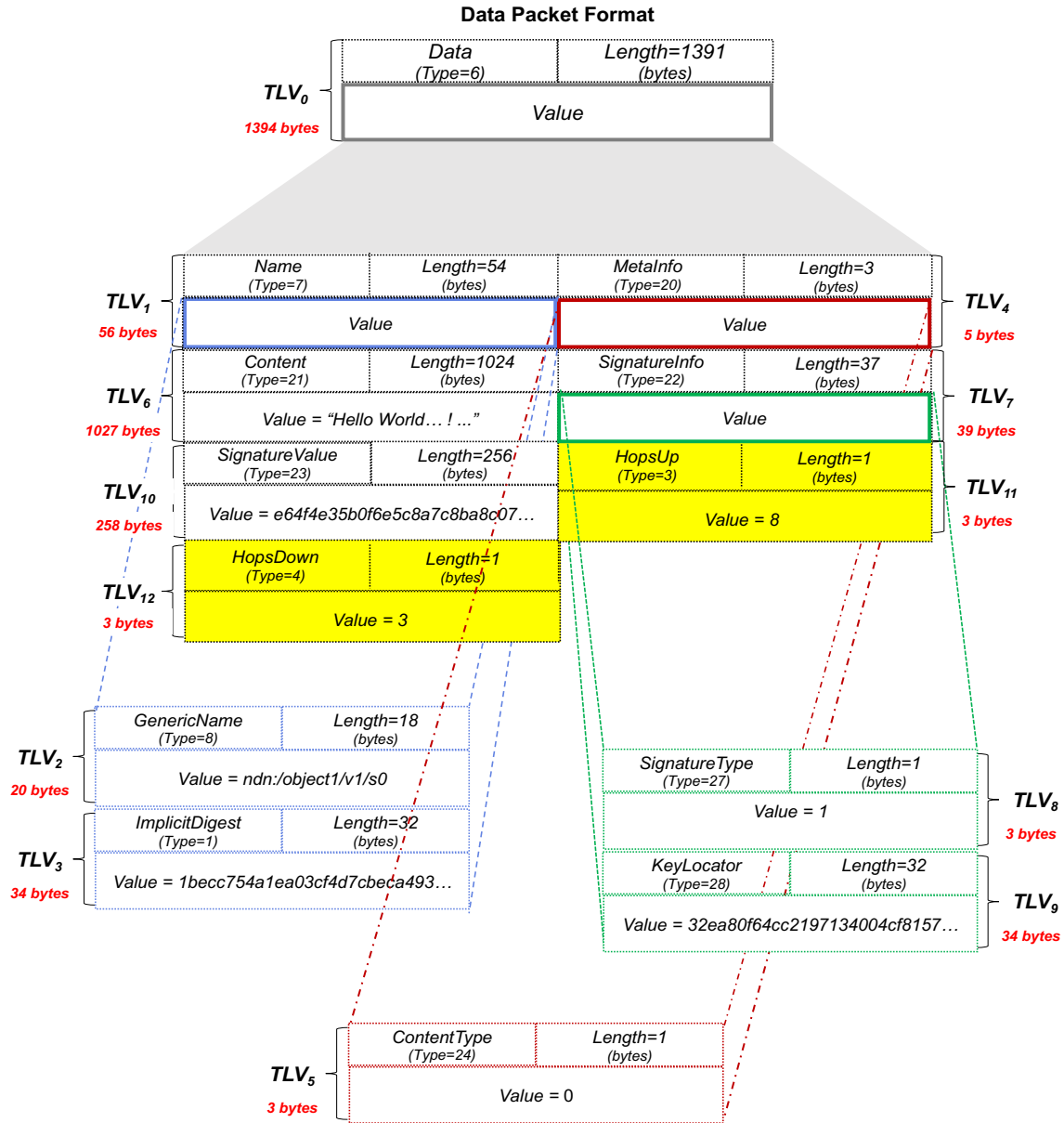


Fig. 3.3: Representation of the fields of a Data packet necessary and/or related to the context of this thesis for the operation of PLbC at an NDN caching-enabled router.

3.2.4 PIT Entry Format

To support the functionality of PLbC, the format of a PIT entry has been modified to consist of the following information. This information is mainly extracted from the Interest packet format upon the arrival of an Interest at an NDN router.

- ***name***: A content identifier used to identify a chunk of an object resource, for which an Interest has been received. This information is extracted from the equivalent field of an Interest.
- ***infaces***: A list of incoming faces through which an Interest(s) for this content identifier has been received since the creation of the entry. The format of a face has been modified to include information with regard to the hopsup field of an aggregated Interest. This information will be used upon the reception of Data to update the hopsup field of each replica to be sent downstream in order to reflect the number of hops that each aggregated Interest would have traversed as if it was propagated upstream to a content source. This way, the downstream routers on delivery paths will be able to calculate the benefit of caching the content locally instead of requesting it from the content source. This type of information is originally hidden due to a operation of a request aggregation mechanism.
- ***outfaces***: A list of outgoing faces through which an Interest(s) for this content has been propagated to a content source(s) since the creation of the entry.
- ***lifetime***: A time counter that once expired, triggers the removal of the entry. A lifetime is initialised and updated upon the arrival of an Interest by extracting the value of the equivalent field from the Interest packet format.
- ***nonces***: A list of nonces of the Interests aggregated by the mechanism of request

aggregation since the creation of the entry. Nonces are used to detect duplicated Interests that have been forwarded through multiple faces within a network and discard them. This information is extracted from the equivalent field of an Interest.

3.2.5 Upstream Process

The term upstream process refers to the arrival of Interests at an NDN router. Following the representation of the Interest and Data packet formats presented in Fig. 3.2 and 3.3, Fig. 3.4 provides a representation of the upstream process at an NDN router as follows:

Upon the arrival of an Interest a router will check if the content identifier of the Interest refers to the first chunk of an object resource. If this holds true, a router will perform an SIT lookup (1). If an SIT entry that matches the prefix of the content identifier of the Interest exists, the popularity counter of this object resource will be updated by increasing its value by 1 (2). If otherwise, an SIT entry will be created (3). Once the SIT lookup is complete and/or if the Interest refers to a chunk other than the first chunk of an object resource, a router will perform a CS lookup (4). If a CS entry that matches the content identifier of the Interest exists, the content will be propagated to the consumer through the face at which the Interest arrived (5), i.e. *face*₂. If otherwise, a router will perform a PIT lookup (6). If a PIT entry that matches the content identifier of the Interest does not exist, a PIT entry will be created (7). Once a PIT entry is created, the hopsup field of the Interest will be updated by increasing its value by 1 (8). The Interest will be then propagated upstream to a content source(s) according to the information in the FIB (9), i.e. *face*₄. If a FIB entry that matches the content identifier of the Interest does not exist, the Interest will be discarded (10). If a PIT entry that matches the content identifier of the Interest exists, the Interest will

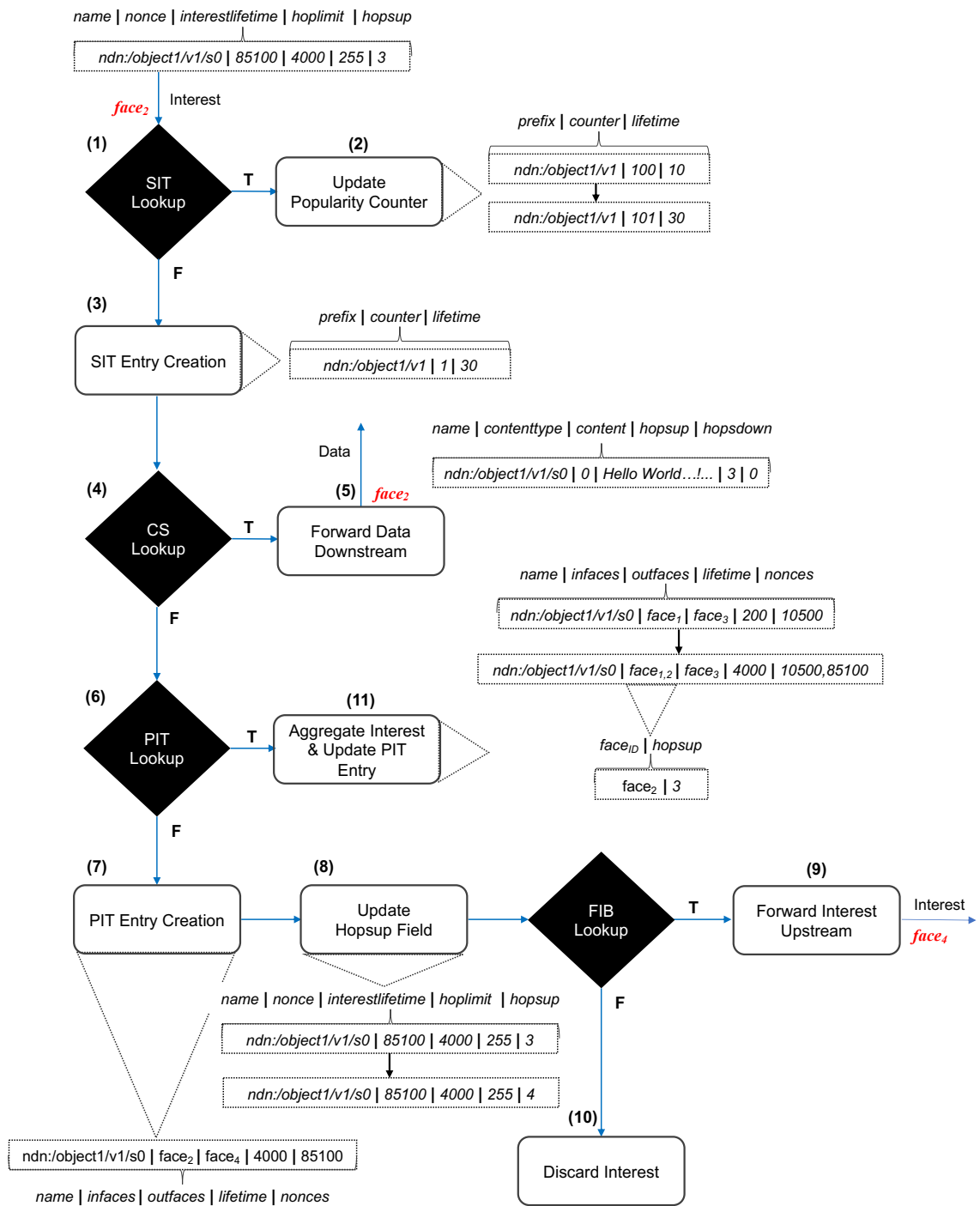


Fig. 3.4: Upstream forwarding process of an Interest at an NDN caching-enabled router that performs caching using the PLbC caching policy.

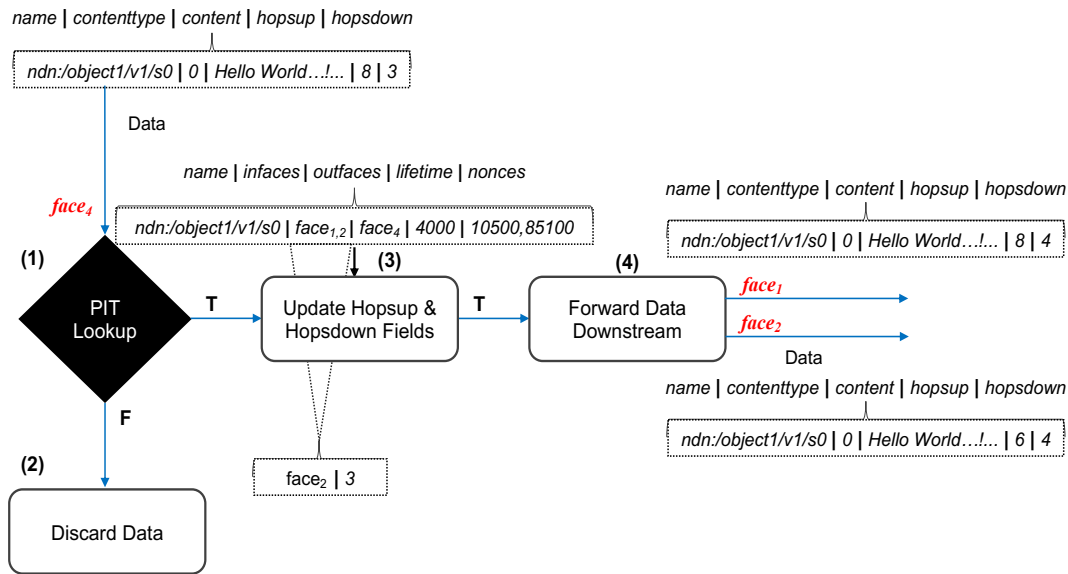


Fig. 3.5: Downstream forwarding process of Data at an NDN caching-enabled router that performs caching using the PLbC caching policy.

be aggregated. The existing PIT entry will be updated according to the information included in the incoming Interest, such as its lifetime and nonce (11). The incoming face on which the Interest has been received will also be added in the list of infaces, including the hopsup metric of the Interest.

3.2.6 Downstream Process

The term downstream process refers to the arrival of Data at an NDN router. Following the representation of the upstream process in Fig. 3.4, Fig. 3.5 provides a representation of the downstream process at an NDN router, described as follows: Upon the arrival of Data a router will perform a PIT lookup (1). If a PIT entry that matches the content identifier of the Interest does not exist, the packet is unsolicited and it will be discarded (2). If otherwise, a replica of the Data will be propagated to each of the consumers

interested in retrieving this content through the faces in the infaces list (4), i.e. $face_1$ and $face_2$. For this purpose, both the hopsup field and the hopsdown field of each replica will be updated before propagating the replicas downstream (3). More specifically, the hopsup field of each Data packet will be updated following the information in the infaces list with regard to the hopsup metric of the Interests, to reflect the number of hops that each aggregated Interest would have traversed as if it was propagated upstream. Similar to the hopsup field of the Interest that was indeed propagated to the content source, the hopsdown field of the Data packet will be updated by increasing its value by 1. After the propagation of Data, the PIT entry will be deleted.

3.3 System Model

In this section, a description of the system model used in the evaluation of PLbC using the ndnSIM(v1) simulator, against the PC+ and CAC caching policies, and the NC approach is given. The comparison of PLbC against PC+ aims to highlight the benefit of PLbC against the caching policy that has been proven to perform the best among the caching policies evaluated in Section 2.3.7. The comparison of PLbC against CAC aims to highlight the benefit of PLbC due to modifying the content popularity factor of CAC into the one adopted in PLbC. The comparison of PLbC against NC aims to highlight the benefit of PLbC with regard to the hop-count metric.

The system model used in the evaluation of PLbC is equivalent to the system model used in the quantitative comparison of caching policies in Section 2.3.7, including the network topologies (Section 2.3.3), applications (Section 2.3.4) and architectural components (Section 2.3.5). Hence, to avoid repetition, the focus of this section lies on highlighting the differences between the two system models. The evaluation metrics

used in the evaluation of PLBC are the ones described in Section 2.3.6.

3.3.1 Applications

Even though no modifications are necessary to the operation of a consumer for the evaluation of PLbC against the alternatives, the operation of a producer, as described in Section 2.3.4.2, has been modified to maintain the value of the hopsup field of an Interest by copying its value into the equivalent field of a Data packet. The use of single-Interest object resources adopted in the evaluation of dynamic-probability caching policies is not expected to affect the performance of PLbC, since PLbC determines the popularity of content on an object naming granularity instead of a chunk naming granularity. Consequently, all chunks of an object resource will have the same popularity.

3.3.2 Architectural Components

Besides the definition of the SIT table, an additional architectural component used to collect information with regard to the popularity of object resources within an ISP/AS network, no further modifications have been applied to the original NDN architecture. Similar to the performance of PLbC, the size of an SIT table is independent of the adaptation of single-Interest object resources, since the popularity counters ctr_o are updated upon the reception of an object request, determined upon the reception of an Interest, and if an Interest refers to the first chunk of an object resource. This is the type of Interests used in the simulations to represent single-Interest object resources.

3.4 Evaluation

In this section, the evaluation results of PLbC against the PC+ and CAC caching policies, and the NC approach, are presented. Similar to the quantitative comparison between a number of caching policies in Section 2.3.7, the evaluation aims to provide valuable information with regard to the effect of the PLbC caching policy on the performance of a network and the experience of consumers. To this end, the evaluation has been conducted using the cache-based metrics of cache-hit ratio (CHR), caching frequency (CF) and cache-eviction ratio (CER), and the network-based metric of hop-count. The first category of metrics is used to quantify the efficacy of a caching policy to cache and maintain the desired content, hence its ability to prevent the propagation of Interests to a content source. The cache-based metrics CF and CER aim to highlight the behaviour of a caching policy upon new content and the utilisation of memory resources at an NDN router. The second category of metrics is used to quantify the performance of a caching policy with regard to the network topology and the reduction of the network traffic in terms of hop-counts traversed by both Interests and Data. Both categories of metrics are used to quantify the benefit of a caching policy to reduce the network traffic within a local ISP/AS network and the traffic propagated beyond the boundaries of the local network to the Internet infrastructure, the main objectives of this thesis.

Besides the evaluation metrics, the same evaluation parameters summarised in Table 2.2 have been used in the evaluation of PLbC. Hence, the limitations described in Section 2.3.9 apply to the evaluation of PLbC as well. For ease of reference, the term *approach* is used to refer to the caching policies and evaluation system. For ease of comparison, the mean and standard deviation (SD) values of the evaluation metrics of an approach are used, with the SD values being presented on top of the mean values

of an evaluation metric. This form of representation of the mean and SD values may exceed the maximum value of an evaluation metric. The SD values are used as a complementary descriptive measurement to the mean values to identify the variance on the behaviour of caching policies and the effect of the evaluation parameters on their performance [Hassani]. Both values have been derived from the evaluation results of 10 simulation runs.

3.4.1 Evaluation Results

Fig. 3.6 and 3.7 illustrate the mean hop-count and SD values of an approach with regard to the catalog size $|O|$ and the cache size cs for the AS-3257 network topology and the AS-3967 network topology, respectively, where $|O| = \{10^6, 10^8\}$ and $cs \in \{10^2, 10^3, 10^4\}$. The mean hop-count of NC is equal to 4 for the AS-3257 network topology and equal to 5 for the AS-3967 network topology. The mean hop-count of CAC is equal to 3 and 4, respectively. The mean hop-count of PC+ is equal to that of CAC when $cs \in \{10^2, 10^3\}$, yet 2 hops lower when $cs = 10^4$ for the AS-3257 network topology. The mean hop-count of PLbC is equal to the mean hop-count of PC+ when $cs = 10^2$ and $cs = 10^4$, yet 1 hop lower when $cs = 10^3$ for the same network topology. The mean hop-count of both PC+ and PLbC is equal to the mean hop-count of CAC when $cs = 10^2$, yet 1 and 3 hops lower when $cs \in \{10^3, 10^4\}$ for the AS-3967 network topology. The SD values of all caching policies are between 1-2 hop-counts for either network topology. Besides the remaining caching policies, whose SD values remain constant as the cache size increases, an increase of 100% of the SD values is shown for PC+ as the cache size increases from $cs = 10^2$ to $cs = 10^3$ for the AS-3257 network topology. Moreover, the SD values of CAC and PC+ indicate a higher variance of the metric for the AS-3967 network topology, compared to the AS-3257 network topology.

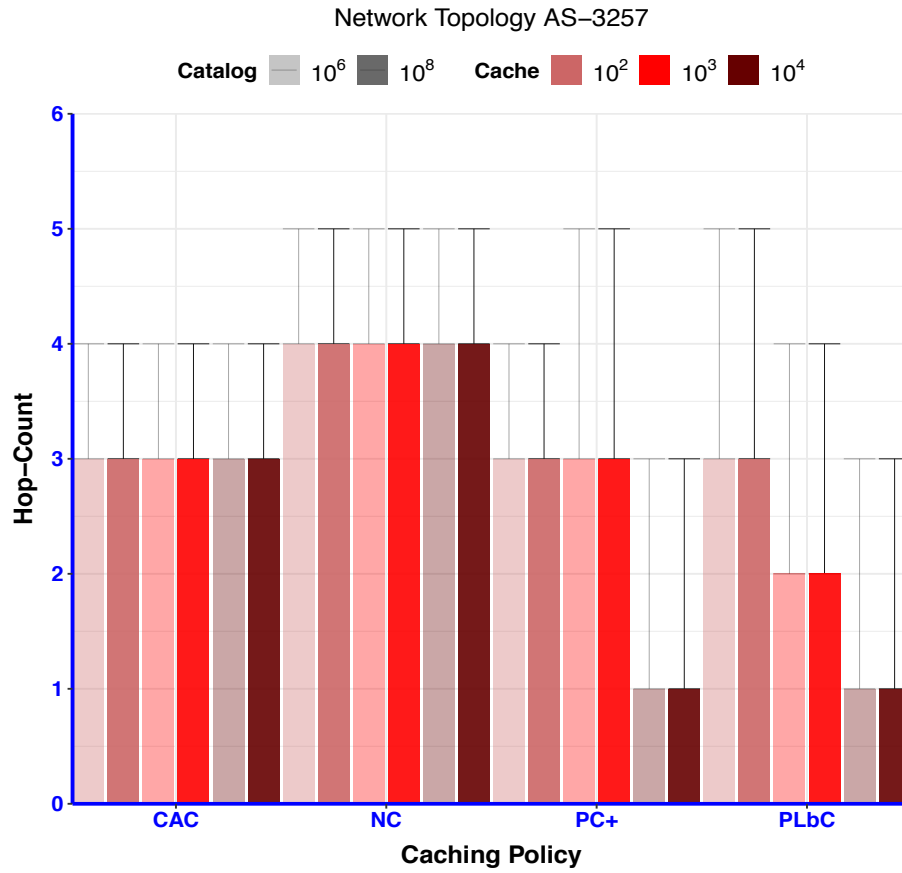


Fig. 3.6: Representation of the mean hop-count and SD values of an approach for the AS-3257 network topology, when $|O| = \{10^6, 10^8\}$ and $cs \in \{10^2, 10^3, 10^4\}$.

According to both figures, and besides the conclusions listed in Section 2.3.7, a number of conclusions may be withdrawn as follows:

- The hop-count metric of PLbC has been shown to be independent from the number of routers on the forwarding and the delivery paths. This claim may be verified by consulting both the mean and SD values of the hop-count metric of PLbC for both network topologies.
- An increase in the cache size of routers has been shown to decrease the hop-count

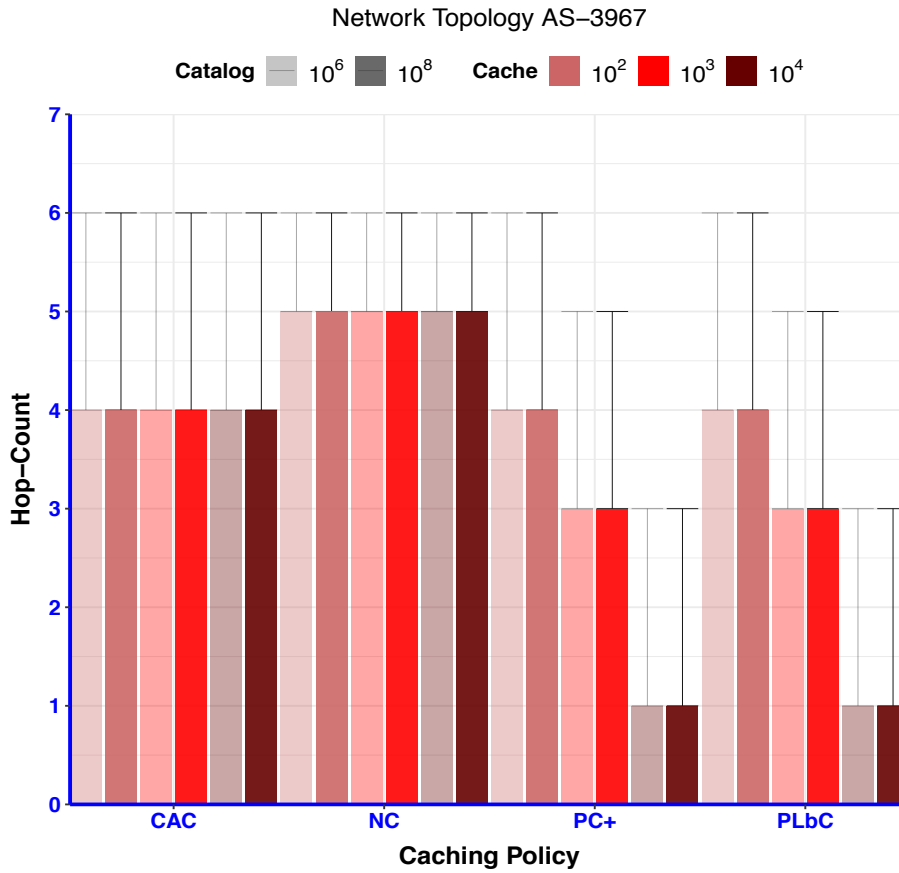


Fig. 3.7: Representation of the mean hop-count and SD values of an approach for the AS-3967 network topology, when $|O| = \{10^6, 10^8\}$ and $csc\{10^2, 10^3, 10^4\}$.

metric of PLbC, while its SD values have been shown to remain constant. This behaviour comes in contrast to the hop-count metric of CAC, who has been shown to remain constant. This difference between the metrics of the two caching policies verifies the claim that the performance of CAC is affected by the content popularity factor and its definition, while highlighting the importance of reforming this factor.

- Similar to the impact of the catalog size on the hop-count metric of a caching policy presented in Section 2.3.7, the hop-count metric of PLbC is independent of

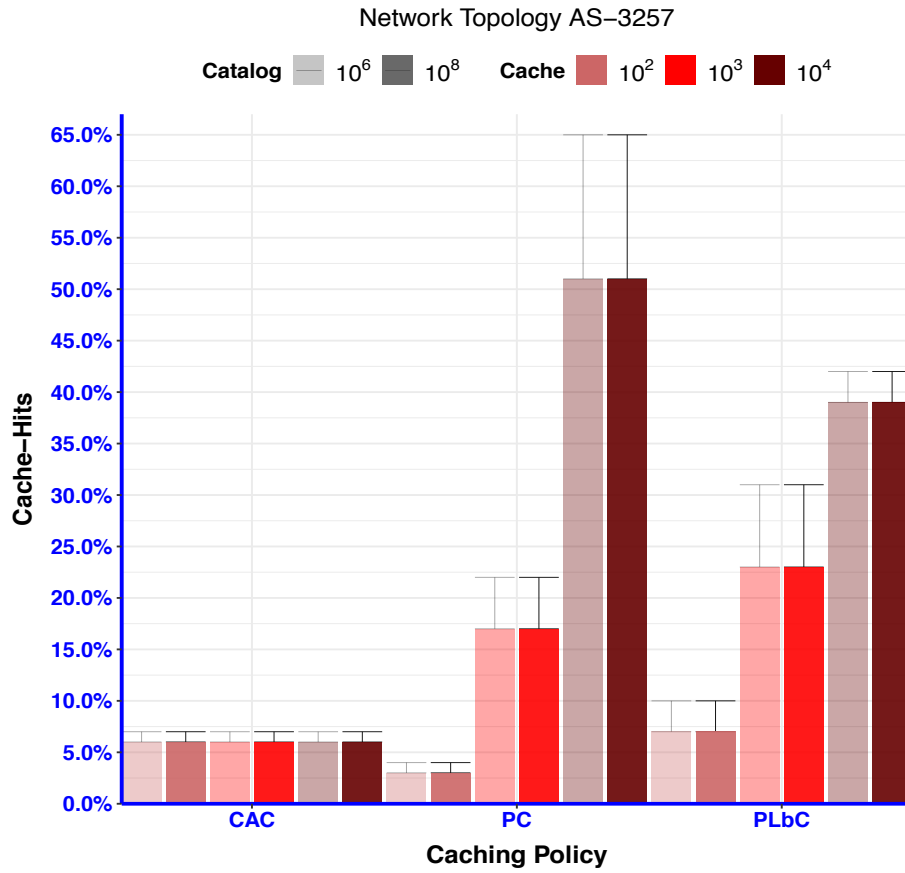


Fig. 3.8: Representation of the mean CHR and SD values of an approach for the AS-3257 network topology, when $|O| = \{10^6, 10^8\}$ and $cs \in \{10^2, 10^3, 10^4\}$.

this parameter.

Fig. 3.8 and 3.9 illustrate the mean CHR and SD values of an approach with regard to the catalog size $|O|$ and the cache size cs for the AS-3257 network topology and the AS-3967 network topology, respectively, where $|O| = \{10^6, 10^8\}$ and $cs \in \{10^2, 10^3, 10^4\}$. The results are equivalent for both network topologies. The mean CHR of CAC is equal to 6% for the AS-3257 network topology, and equal to 4% and 5% for the AS-3967 network topology, when $cs = 10^2$ and $cs \in \{10^3, 10^4\}$, respectively. These values are 2-

3% higher than the mean CHR of PC+ when $cs = 10^2$, for either network topology. In contrast to CAC, the mean CHR of the remaining caching policies increases as the cache size increases, with PLbC outperforming PC+ when $cs = \{10^2, 10^3\}$, and PC+ outperforming PLbC when $cs = 10^4$. The mean CHR of PLbC is equal to 7%, 23% and 39% when $cs = \{10^2, 10^3, 10^4\}$ for the AS-3257 network topology. The equivalent CHR values of PC+ are equal to 3%, 17% and 51%, respectively. The mean CHR of PLbC is equal to 6%, 21% and 38% when when $cs = \{10^2, 10^3, 10^4\}$ for the AS-3967 network topology. The equivalent CHR values of PC+ are equal to 3%, 16% and 48%, respectively. Besides CAC, whose SD values remain constant as the cache size increases, the SD values of PC+ are between 1-15%, while a higher SD value is shown as the cache size increases for either network topology. Interestingly, the SD values of PLbC are equal to 3-4% when $cs \in \{10^2, 10^4\}$ and equal to 8% when $cs = 10^3$ for either network topology, indicating a more stable behaviour of PLbC for lower and higher cache sizes.

According to both figures, and besides the conclusions listed in Section 2.3.7, a number of conclusions may be withdrawn as follows:

- Similar to the impact of the cache size and the catalog size on the hop-count metric of PLbC, an equivalent impact is observed for the CHR metric, while an interesting difference is the impact of the cache size on the SD values of PLbC.
- The CHR metric of PLbC decreases, while its SD value increases, as the number of routers on the forwarding and the delivery paths increases. This is equivalent to the impact of the same parameter on the CHR metric of HPC and PC+. A higher number of routers on the forwarding and the delivery paths decreases the caching probability on delivery paths, causing a reduction in the mean CHR values.
- Similar to the impact of the content popularity factor of CAC when $cs = \{10^3, 10^4\}$

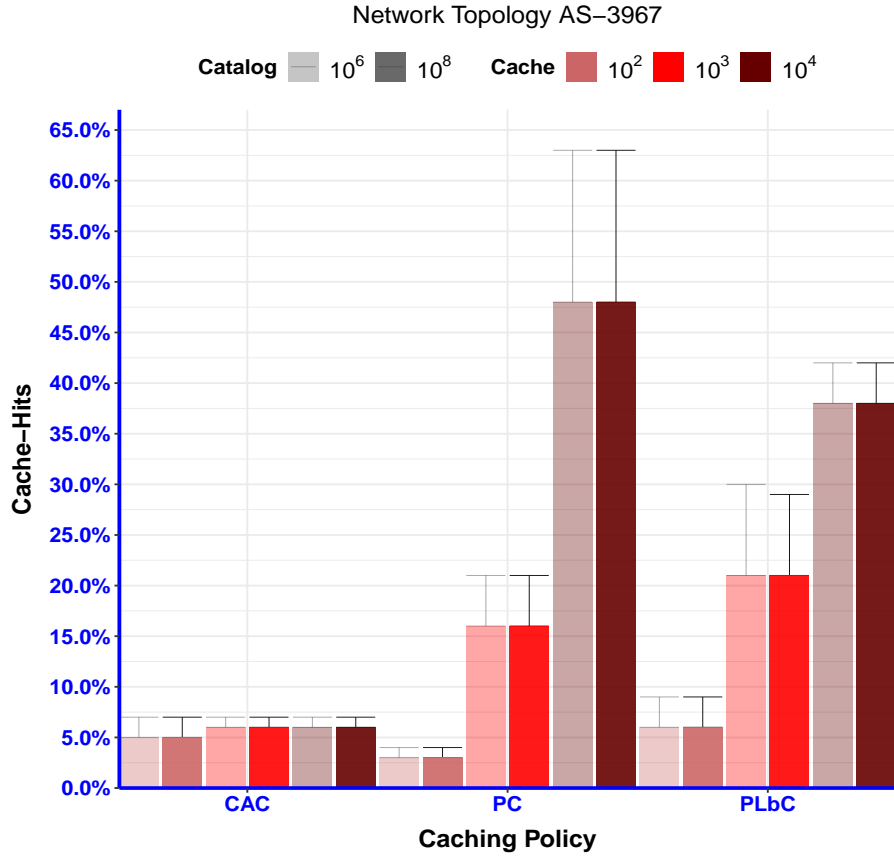


Fig. 3.9: Representation of the mean CHR and SD values of an approach for the AS-3967 network topology, when $|O| = \{10^6, 10^8\}$ and $cs \in \{10^2, 10^3, 10^4\}$.

and the underutilisation of the cache capacity of routers, an equivalent impact has been observed for the CHR metric of PLbC when $cs = 10^4$. Nonetheless, this has not shown to affect the performance of PLbC with regard to the hop-count metric.

Fig. 3.10 and 3.11 illustrate the mean CF and SD values of an approach with regard to the catalog size $|O|$ and the cache size cs for the AS-3257 network topology and the AS-3967 network topology, respectively, where $|O| = \{10^6, 10^8\}$ and $cs \in \{10^2, 10^3, 10^4\}$. Referring to both figures, the following observations may be made:

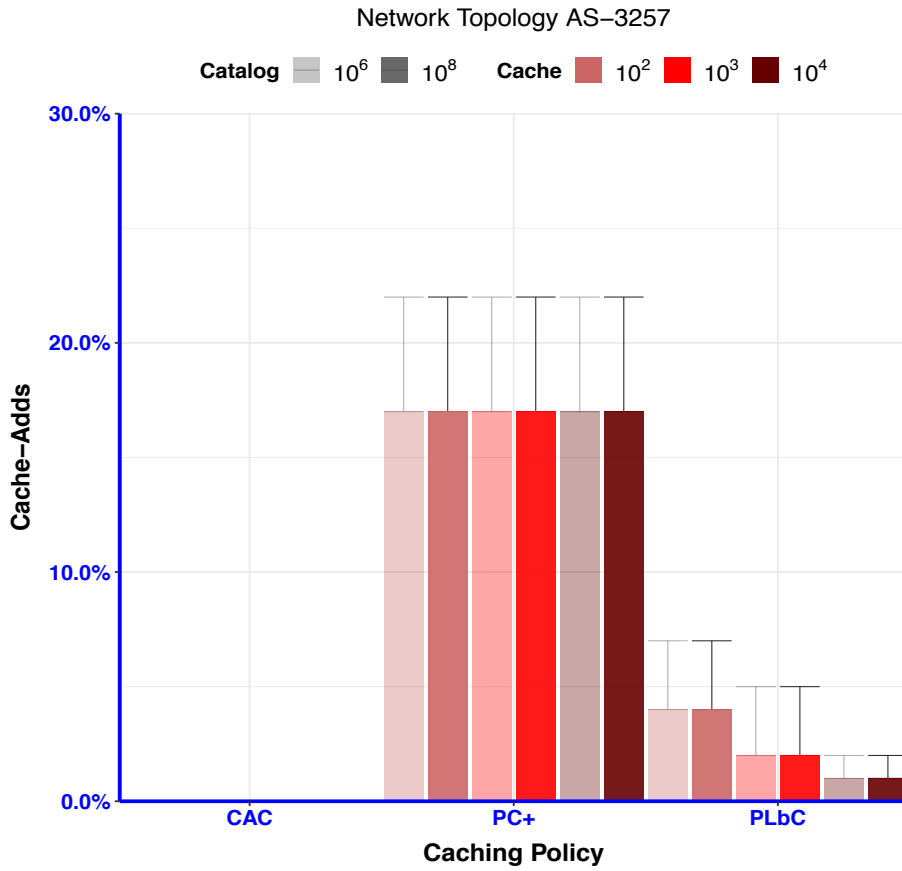


Fig. 3.10: Representation of the mean CF and SD values of an approach for the AS-3257 network topology, when $|O| = \{10^6, 10^8\}$ and $cse\{10^2, 10^3, 10^4\}$.

- The mean CF values of both CAC and PC+ are independent of the parameter of cache size, yet dependent on the parameter of the network topology.
- The mean CF values of PLbC depend on both evaluation parameters. PLbC results in lower CF values as the cache size increases, while its SD values range between 1-4%. The mean CF of PLbC is equal to 4%, 2% and 1% when $cse\{10^2, 10^3, 10^4\}$ for the AS-3257 network topology. The equivalent CF values of PLbC are 1% higher when $cse\{10^2, 10^3\}$ and equal if otherwise for the AS-3967 network topology.

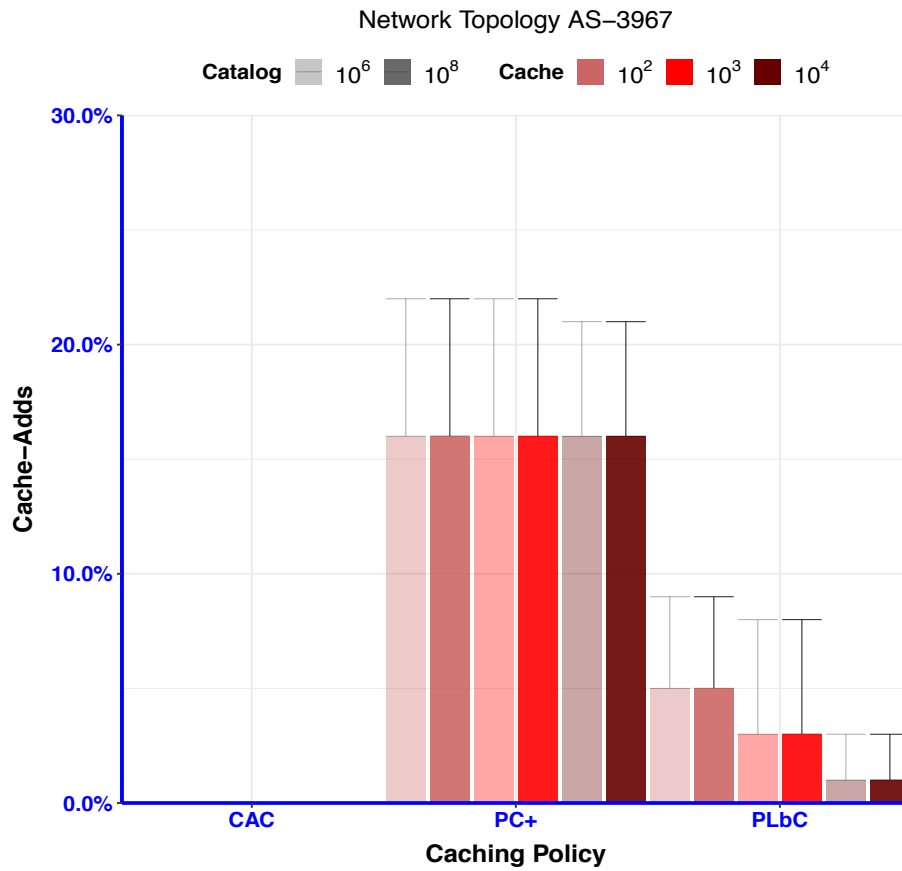


Fig. 3.11: Representation of the mean CF and SD values of an approach for the AS-3967 network topology, when $|O| = \{10^6, 10^8\}$ and $c\epsilon \in \{10^2, 10^3, 10^4\}$.

- The difference of the cache size on the CHR metric of PLbC and PC+ may be explained as follows: PLbC is a caching policy whose functionality is based on the content popularity factor. Hence, PLbC is able to cache and also maintain a higher amount of content in routers as the cache size increases. This way, the necessity to cache new content is eliminated. This claim may be verified by consulting Fig. 3.12 and 3.13 which illustrate the mean CER and SD values of an approach with regard to the evaluation parameters considered. Consulting the results, the mean CER of

PLbC is 1%, 18% and 70% lower to that of PC+ when $cs \in \{10^2, 10^3, 10^4\}$ for the AS-3257 network topology. The mean CER of PLbC is 1%, 14% and 65% lower to that of PC+ for the AS-3967 network topology and the same cache sizes. The SD values of PC+ range between 0-11%. The SD values of PLbC range between 1-30%, indicating a higher variance of the metric. A higher SD value is shown for both caching policies as the cache size increases.

- The mean CF of CAC is equal to 0% for either network topology; the CF metric of CAC has been explained in Section 2.3.7. The SD values of CAC are equal to 0%, since the mean values of the CF metric of CAC are equal to 0%.
- The mean CF of PC+ is equal to 17% for the AS-3257 network topology and equal to 16% for the AS-3967 network topology. The SD values of PC+ range between 5-6%, depending on the cache size.
- An increase of up to 2% of the SD values of the caching policies is shown for the AS-3967 network topology, compared to the AS-3257 network topology, following the mean values.

3.5 Discussion

The purpose of this section is to provide a discussion on the effect of the caching criteria of PLbC on the performance of a caching policy and the network, and the experience of consumers. Moreover, the discussion aims to conclude to the approach that yields to the highest performance gain among the approaches evaluated in Sections 2.3.7 and 3.4.1. In addition to this, this section aims to highlight the limitations derived from the design and the evaluation of PLbC.

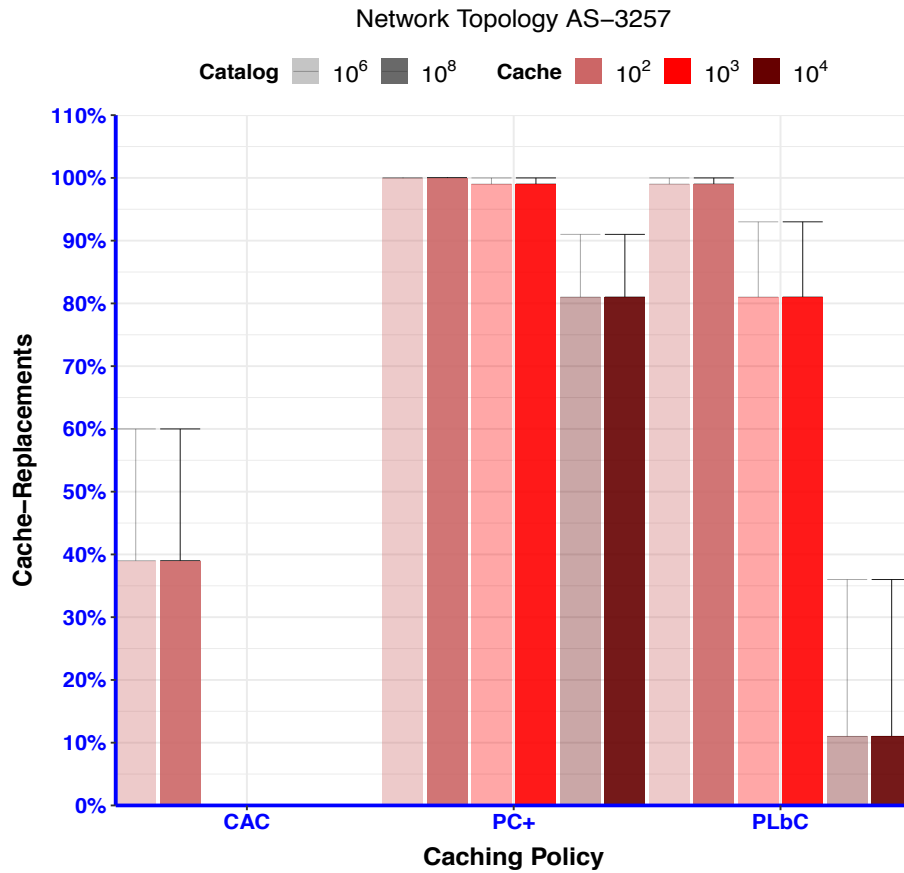


Fig. 3.12: Representation of the mean CER and SD values of an approach for the AS-3257 network topology, when $|O| = \{10^6, 10^8\}$ and $cs \in \{10^2, 10^3, 10^4\}$.

3.5.1 Lessons Learned

In addition to the summary of the performance of the PC+, HPC, CAC, LCE and LCL caching policies presented in Section 2.3.8, a number of conclusions may be derived for the performance of PLbC and the caching criteria used, summarised as follows:

- PLbC is a caching policy whose functionality is based on the criterion of content popularity, designed based on the design of CAC adopted in Section 2.3.2.3, and the criterion of the location of routers on delivery paths. PLbC yields a higher per-

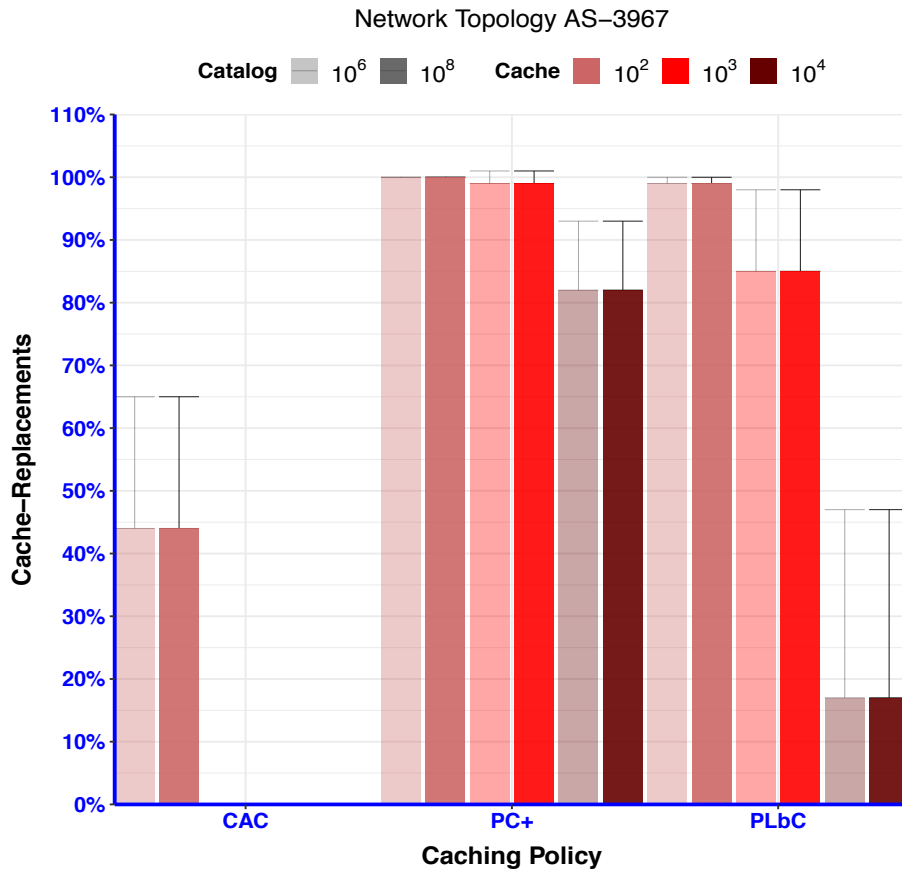


Fig. 3.13: Representation of the mean CER and SD values of an approach for the AS-3967 network topology, when $|O| = \{10^6, 10^8\}$ and $csc\{10^2, 10^3, 10^4\}$.

formance gain than CAC for either evaluation metric, highlighting the limitations caused by the content popularity factor and the benefit of reforming this factor.

- Similar to the effect of the cache size on the performance of a caching policy proven in Section 2.3.7, an increase in the cache size of routers has been shown to benefit the performance of PLbC with regard to either evaluation metric.
- PLbC is unable to fully exploit the cache capacity of routers when $cs = 10^4$. This claim may be verified by consulting the results for the CHR metric. In contrast to

CAC, this limitation has not been shown to affect the hop-count metric of PLbC. At this point it is worth noting, that due to the adaptation of single-Interest object resources, a cache size of $cs = 10^4$ chunks used in simulations is equivalent to a cache size of 100GBs in real deployments. This size is 10 times higher to the one suggested by the NDN research community to bound the CS lookup times. Hence, although this conclusion is alarming, its application is questionable given the computational overhead that such cache sizes are expected to introduce.

- In contrast to the remaining caching policies whose performance has been shown to be affected by the network topology parameter for the hop-count metric, the performance of PLbC has been shown to be more robust.
- In contrast to the remaining caching policies, PLbC is able to cache, and also maintain content that satisfies a higher number of requests. This claim may be verified by comparing the results for both PC+ and PLbC for the CHR and the CF metrics. More specifically, PLbC yields a 3-6% higher probability of retrieving the content locally from the caches of routers compared to PC+ when $cs = \{10^2, 10^3\}$, while caching 11-15% less content at the routers within a network.
- PLbC performs either equivalent or better to PC+. This claim may be verified by consulting the evaluation metrics of hop-count, CHR and CF. An exception to this is when the cache size of routers $cs = 10^4$, with regard to the CHR metric. Nonetheless, as already mentioned the application of this scenario is questionable given the computational overhead that such cache sizes are expected to introduce.

The conclusions presented in this section highlight the superiority of PLbC and the benefits derived from its functionality and the caching criteria used.

3.5.2 Evaluation Limitations

Since the system model adopted in the evaluation of PLbC is equivalent to the system model adopted in the quantitative comparison of a number of caching policies presented in Section 2.3.7, any assumptions and limitations highlighted in Sections 2.3.4 and 2.3.9, apply to this system model as well.

Taking into consideration the limitations encountered in the evaluation of full object resources using the ndnSIM simulator, the evaluation of PLbC has been limited to the AS-3257 network topology and the AS-3967 network-topology provided by the ndnSIM community for experimentation. The size of either network topology is less than 100 routers. Such network topologies may be safely used to provide us with an intuition about the performance of a caching policy. Nonetheless, the results concluded may differ in real deployments, where the values of the evaluation parameters considered may differ as well. Similar to the results presented in Section 2.3.7, an increase in the number of routers within a network is expected to decrease the CHR metric of PLbC. This claim may be verified by consulting the results of PLbC for the AS-3257 network topology, against the AS-3967 network topology. Moreover, an increase in the number of routers within a network is expected to increase the number of routers on the forwarding and the delivery paths, which in turn will affect the P_l factor of PLbC. Since the hop-count metric is independent of the link capacity and the delay introduced per link, a lower hop-count metric is expected compared to the alternative caching policies. This claim may be verified by the fact that PLbC is a caching policy that aims to cache popular content closer to consumers. Nonetheless, a lower hop-count metric does not guarantee lower content delivery times for consumers in real deployments, since the content delivery metric depends on both the capacity and the delay of the links traversed. Consulting

Section 2.3.9, heterogeneous network topologies, where the edge routers within a network are connected to lower capacity links than the backbone routers, are expected to impact the content delivery times of consumers concluded by PLbC. The amount of this impact strongly depends on the location of the content. As an example, a higher benefit of the PLbC caching policy is expected as the number of edge links to be traversed by both Interests and Data increases, since a higher delay is expected to be introduced to the content delivery times of consumers compared to the backbone links. In addition to the aforementioned parameters, the connectivity of a network is also expected to affect the performance of PLbC. More specifically, highly connected network topologies, such as the AS-3257 network topology and the AS-3967 network topology, benefit all cache-based metrics since a higher number of content requests is expected to be propagated through the same set of routers on the forwarding and the delivery paths.

In addition to the impact of the network topology parameter on the performance of PLbC, the popularity distribution adopted by consumers to issue object requests is also expected to affect its performance, since content popularity distributions define the benefit of caching a particular content [Carofiglio a, Rossi a]. Taking into account the popularity distributions that may be adopted in the evaluation of caching policies, such as a Zipf distribution used to generate Web traffic or a Zipf-Mandelbrot distribution used to generate P2P traffic [Pentikousis], a modification of the results presented in Section 3.4.1 may be concluded. More specifically, highly skewed distributions are expected to highlight the benefits of PLbC since a higher number of object requests is expected to be issued for a subset of objects resources, easing the functionality of PLbC to cache and maintain popular content. On the contrary, low skewed distributions are expected to degrade its performance since object requests will be more scattered.

Similar to the discussion on the impact of single-Interest object resources on the

performance of a caching policy in Section 2.3.9, single-Interest object resources are not expected to affect the performance of PLbC, since neither its functionality nor the evaluation metrics adopted are affected by this difference. Nonetheless, the adaptation of full object resources is expected to highlight the differences on the performance of PLbC against the alternatives, since the same evaluation results are expected to be applied to more contents, i.e. 10^3 more contents per object resource on average.

Another evaluation parameter expected to affect the performance of PLbC is the catalog size. A higher catalog size is expected to highlight the benefit of PLbC to cache and maintain popular content. This claim may be verified based on the fact that a higher number of contents is expected to be competing for the same cache entry, reducing the fraction of contents that may be cached within a network. Yet, consulting Section 3.4.1, PLbC has been shown to yield a higher performance benefit against the alternatives when the cache capacity of a network is much smaller than the amount of content available.

3.5.3 Design Limitations

Referring to sections 3.4.1 and 3.5.1, PLbC has been shown to outperform the caching policy proven to perform the best among the caching policies evaluated in chapter 2, i.e. PC+. Nonetheless, PLbC has been shown to be unable to fully utilise the cache capacity of routers cs when $cs = 10^4$, with regard to the CHR metric. Similar to CAC, the reason for this limitation is the design of the popularity factor P_f , as described in Section 3.2.2. Although the design of the P_f factor adopted in PLbC has been shown to perform satisfactory overall, a modification may be applied to further exploit the cache capacity of routers cs within a network when this increases. Similar to the design of PC+ that utilises information with regard to the cache capacity of routers on delivery paths, PLbC could be modified to take into account an equivalent parameter.

More into the design of PLbC and its limitations, an additional architectural component, i.e. the SIT table, is required to maintain the popularity counters ctr_o of object resources used in the calculation of the caching probability p . The use of an SIT table is a necessary prerequisite to the design of caching policies that utilise correlativity-based popularity calculations [Badov , Janaszka]. The size of this table is expected to grow linearly with the number of individual objects requested within an ISP/AS network. Following the examples of SIT entries given in Fig. 3.1, the size of an SIT entry is assumed to be equal to 25 bytes. Following this convention and the catalog sizes suggested for UGC content including one-timer objects, i.e. 10^8 object resources, the size of an SIT table is calculated to be equal to 2.5GB at maximum. To maintain the size of SIT tables to a minimum, an SIT table collects statistical information about object resources for which the requesting activity remains active within the local ISP/AS network, allowing this way old information to expire and one-timer object counters to be removed from the table. For this purpose a lifetime is bound to an SIT entry that triggers the removal of the entry. Although a number of suggestions have been presented in Section 3.2.1 for the initialisation of a lifetime for UGC content, a comprehensive examination of the impact of a lifetime on the performance of PLbC has been defined as future work.

3.5.4 Security Implications

Security in NDN is focused on securing the content itself, instead of securing the connection between two endpoints. This design convention allows content to be freely distributed within the Internet infrastructure and to benefit from the mechanisms build at the network layer of the architecture, such as on-path caching. For this purpose, each piece of content is individually signed by its producer/publisher upon publishing the content using a cryptographic public-private key pair, making the content verifiable and

optionally confidential [Zhang b]. Referring to Section 2.1.2.3, a Data packet consists of the signatureinfo and signaturevalue fields. Combining these fields allows consumers to ensure the integrity and/or provenance verification of the content contained within a Data packet upon reception. The signatureinfo field consists of the SignatureType and KeyLocator fields. The SignatureType field is used to maintain information about the type of the signature algorithm, e.g. an SHA-256 algorithm. The KeyLocator field is used to identify another Data packet that contains a certificate or a public key used to verify the content of this Data packet, or a KeyDigest used to identify the public key, assuming that a trust model exists [Yu b]. This way, consumers may obtain the public key bound to the content received to ensure the integrity, origin authentication and correctness of the content.

The security implications derived from the design of PLbC are not content-related. The mechanisms described above have been included to highlight the security steps that need to be considered to ensure that a communication within an ISP/AS network over the NDN architecture that utilises the PLbC caching policy is secure. The security implications derived from the design of PLbC refer to the integrity of the Interest and Data packet formats and the raise of potential *cache pollution attacks* [Mannes]. More specifically, either the Interest or Data packet fields related to the functionality of PLbC could be modified, either by an intermediate adversary or a malicious router, to enforce suboptimal caching decisions at the downstream routers on delivery path(s). This is similar to the *Interest Flooding Attack (IFA)* defined in NDN, a type of DoS attack under which an adversary generates a large number of close-in-time successive Interests with the aim to overflow the PIT [Gasti]. Once a PIT is overflowed, it either rejects incoming Interests or deletes PIT entries. To overcome this defect, a number of security mechanisms that detect and mitigate the occurrence of an IFA attack have been

proposed, where routers monitor the rates of outstanding Interests, or the amount of incoming Interests per face [Afanasyev c, Compagno ,Dai].

Although an IFA attack may result in DoS, a cache pollution attack may at worst result in a performance degradation of the network and the experience of consumers. Towards ensuring that the fields of either packet format will remain intact, a lightweight hop-by-hop encryption algorithm, e.g. an SHA algorithm, could be used. Nonetheless, an SHA algorithm is not sufficient to ensure the integrity of a Data packet, while an extra computational overhead is expected to be introduced at intermediate routers. To this end, alternative mechanisms may be used to identify whether an Interest and/or a Data packet have been modified to trigger suboptimal caching decisions. As an example, routers are expected to collect statistical information with regard to their cache performance. This information may even be collected by an NMM within a local ISP/AS network to detect malicious routers. In this context, a cache pollution attack may be detected by comparing the cache performance rates of a router against the equivalent rates of this router in consecutive time intervals. If these rates have changed significantly, a potential cache pollution attack may be in progress. To further identify the origin of this attack, a router may monitor the cache performance rates between consecutive time intervals in relation to the faces through which the content was arrived. Once the potential malicious faces are identified, a filter may be set in the FIB of the router to avoid using the corresponding face/link. Moreover, a push-back alert may be issued to upstream routers that a potential attack may be in progress. This push-back technique may also be used to detect and avoid malicious routers on delivery paths [Gasti ,Yi a]. Similar approaches to detect and mitigate cache pollution attacks have been proposed in the existing literature [Conti ,Karami ,Salah ,Xie ,Xu]

3.6 Summary

In this chapter, a dynamic-probability caching policy, called Popularity and Location-based Caching (PLbC) policy, has been proposed. PLbC aims to selectively cache content that is expected to satisfy a higher number of Interests closer to consumers. This way, PLbC aims to reduce the number of hops to be traversed within a network and the network traffic generated due to the propagation of Interests and Data. To achieve its goal, PLbC utilises both the criterion of content popularity and the criterion of the location of routers on delivery paths, by incorporating them into the caching decision process to construct a probability. To determine which content to cache, PLbC utilises the information collected by the Statistics Information Table (SIT) with regard to the object requests received at an NDN router. To determine at which router to cache content, PLbC utilises the information collected in the Interest and Data packets with regard to the number of hops that each of them has traversed.

PLbC has been evaluated against the PC+ and CAC caching policies, and the NC approach, using the evaluation metrics of hop-count, CHR, CF and CER, and a number of evaluation parameters. According to the results, PLbC has been shown to outperform the alternatives with regard to either evaluation metric. Nonetheless, PLbC has been shown to underutilise the cache capacity of routers when $cs = 10^4$, resulting in a lower performance gain compared to PC+. Even though, this conclusion is alarming for the performance of PLbC, its application is questionable given the fact that a cache size of $cs = 10^4$ used in the simulations is equal to a cache size of 100GBs in real deployments. Besides the benefits of PLbC, a limitation worth further investigation is the maintenance of the SIT table and the effect of the size of this table on the performance of PLbC.

Chapter 4

CS-ERA: A Mechanism for Content Sharing and Inter-network Traffic Reduction

In this chapter, a mechanism that aims to exploit the request aggregation mechanism of NDN to reduce the traffic within a local ISP/AS network, and beyond the boundaries of this network to the Internet infrastructure is proposed, called *Content Sharing-Extended Request Aggregation (CS-ERA)*. By aggregating successive Interests and by bounding these Interests to be satisfied locally within an ISP/AS network by consumers that possess a replica, CS-ERA is expected to reduce the inter-network traffic and the content delivery times of consumers, thus satisfying both the objectives of this thesis.

CS-ERA focuses on content sharing between consumers that request to retrieve the same object resource. For this purpose, CS-ERA exploits the mechanism of request aggregation to collect information with regard to the consumers interested in the same

object resource, and exploit this information to bound successive Interests to be satisfied within the boundaries of the local ISP/AS network, if a local replica exists. To increase the number of Interests to be bounded, CS-ERA extends the naming granularity to which request aggregation is applied from a chunk to an object resource, allowing any Interest that refers to either chunk of an object resource to be aggregated, instead of the Interests that refer to a single chunk of an object resource. This way, CS-ERA increases the probability of request aggregation occurring within a NDN network. This is achieved similar to a number of on-path caching policies (Section 2.2.4), by modifying the Interest and Data packets to include information that will allow the communication between consumers within the same ISP/AS network and the satisfaction of Interests. To propagate Interests to a replica within the local ISP/AS network, an extended Best Route (BR) forwarding strategy has been adopted, called *Locality-First Best Route (LFBR)*.

To further increase the probability of request aggregation occurring, CS-ERA is applied at the edge routers of an ISP/AS network that propagate traffic to the Internet infrastructure. This type of edge routers is different to the one introduced in Section 2.2.1, connected to consumers. Therefore, for the rest of this thesis and to avoid any misconceptions, the term *border router* is used. The term *CS-ERA border router* is used to refer to the border routers that CS-ERA is applied. The rationale behind the selection of border routers for the operation of CS-ERA lies in the fact that border routers are more likely to receive successive Interests that refer to the same object resource since they act as rendezvous points for the traffic to be propagated. Hence, border routers are more likely to increase the probability of request aggregation occurring within a network, while being part of the local network infrastructure. The rest of the routers within an NDN network, i.e. both the edge routers and the intermediate routers, are assumed to operate using the original request aggregation mechanism.

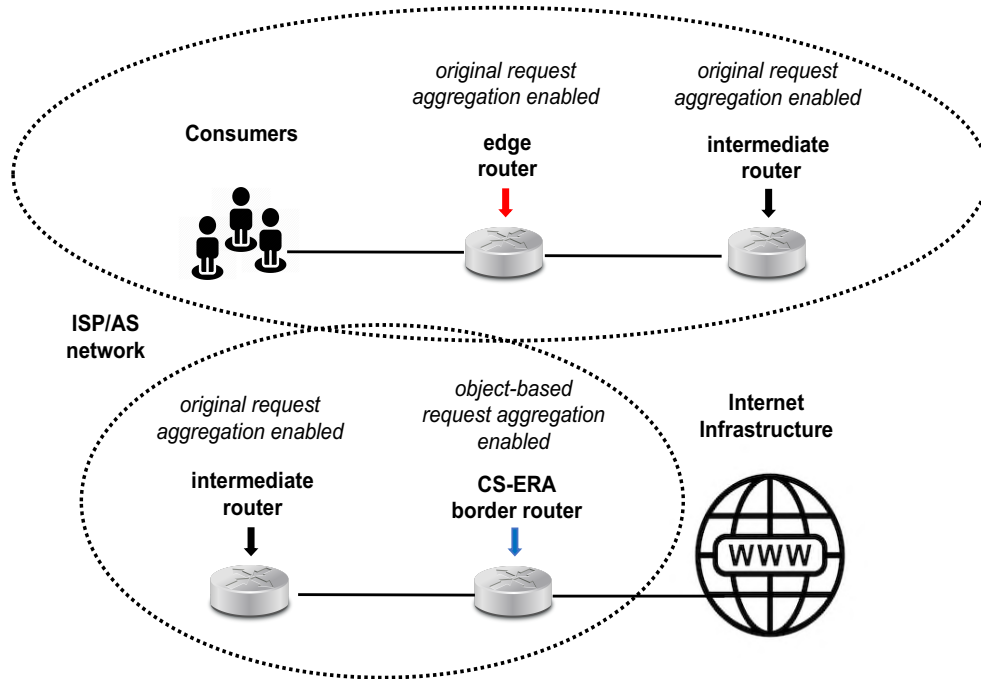


Fig. 4.1: Abstract representation of the functionality of an edge router, an intermediate router and a CS-ERA border router within an ISP/AS network. All routers perform caching on a chunk naming granularity, as defined by the NDN architecture.

4.1 Context and Motivation

Recalling Section 2.4 and the analysis of the efficiency of a request aggregation mechanism, two factors have been identified to be critical to the performance of request aggregation, summarised as follows:

- The time interval for which an entry remains alive in the PIT, while a successive Interest for the same chunk of an object resource is received at an NDN router.
- The efficiency of the on-path caching policy applied to an NDN router.

By consulting the aforementioned observations, it is clear that the interaction of a

request aggregation mechanism with a caching policy, or its dependency on the lifetime of PIT entries can not be changed. Consequently, the aim of this thesis to enhance the performance of CS-ERA through enhancing the efficiency of the request aggregation mechanism of NDN, is neither to disable the feature of on-path caching, nor to increase the time for which entries remain alive in the PIT. Although disabling the feature of on-path caching is expected to increase the efficiency of request aggregation [Dabirmoghaddam, Ghali], different studies have suggested that request aggregation may be complementary to on-path caching, increasing the overall network performance [Shannigrahi]. This contradiction depends on the chunk size of the traffic model adopted, i.e. equal to 100MB, compared to the default chunk size defined, i.e. equal to 10KB [Muscarillo, Carofiglio]. This increase in the chunk size suggests an increase on the lifetime of PIT entries by a magnitude of 10^4 , which in turn increases the probability of request aggregation occurring within an NDN network. Nonetheless, an increase on the lifetime of PIT entries also suggests an increase on the size of a PIT, raising both scalability and deployability concerns (Section 1.3.1).

To avoid the complications associated with the increased PIT entry lifetimes and PIT sizes, the focus of this thesis lies on the exploration of alternatives to increase the probability of request aggregation occurring within an NDN network, when the chunk size of an object resource is equal to 10KB. To this end, the scale on which request aggregation is applied to an NDN router has been extended from a chunk to an object naming granularity. The rationale behind this choice is that a request aggregation mechanism based on an object naming granularity will allow multiple chunks of an object resource to be merged by the same PIT entry instead of a single chunk alone.

4.2 Design and Implementation

In this section, a description of the design of CS-ERA, including the modifications applied to the original request aggregation mechanism and the NDN architecture, as described in Section 2.1, is presented. For ease of understanding, Figure 4.1 provides an abstract representation of the functionality of an edge router, an intermediate router and a CS-ERA border router within an ISP/AS network. All routers perform caching on a chunk naming granularity, as defined by the NDN architecture.

4.2.1 CS-ERA Border Router

CS-ERA is applied to the border routers of an ISP/AS network, i.e. the edge routers connected to the Internet infrastructure. This way, CS-ERA is expected to increase the number of Interests received at border routers, thus increase the probability of request aggregation occurring within an NDN network. By doing so, CS-ERA is expected to increase the number of Interests bounded to be satisfied within the boundaries of the local ISP/AS network, if a local replica exists. The rationale behind the selection of border routers is no different to the functionality of location-based caching policies, where a set of routers within a network may be chosen based on a centrality metric to perform on-path caching, assuming that a higher number of Interests will be satisfied at these routers. Similar, CS-ERA has been chosen to be applied to the edge routers of an NDN network included in the sets of shortest paths between the consumers within a network and a content source, a form of BC metric (Section 2.2.1). The remaining routers within an NDN network operate using the original request aggregation mechanism. For ease of understanding, Algorithm 2 provides the pseudocode at a CS-ERA border router. The algorithm is tightly coupled with the information provided in the following sections.

Algorithm 2 Functionality of CS-ERA mechanism

Require: CS-ERA border router r

- 1: Read $prefix, seq$ from $Interest$ at r
 - 2: $PITentry \leftarrow prefix$ PIT lookup at r
 - 3: **if** $PITentry == 0$ **then**
 - 4: Create $PITentry$ at r
 - 5: Send $Interest$ upstream
 - 6: **else if** $Interest\ seq == PITentry\ seq$ **then**
 - 7: Apply original request aggregation
 - 8: **else if** $Interest\ seq \leq PITentry\ seq$ **then**
 - 9: $PITentry\ nonces \leftarrow PITentry\ nonces + Interest\ nonce$
 - 10: $PITentry\ lifetime \leftarrow Interest\ lifetime$
 - 11: Create $InterestNACK$ at r
 - 12: $InterestNACK\ prefix \leftarrow Interest\ prefix$
 - 13: $InterestNACK\ seq \leftarrow Interest\ seq$
 - 14: Send $InterestNACK$ downstream via the face where the $Interest$ arrived at r
 - 15: **else if** $Interest\ seq \geq PITentry\ seq$ **then**
 - 16: Create $InterestNACK$
 - 17: $InterestNACK\ prefix \leftarrow PITentry\ prefix$
 - 18: $InterestNACK\ seq \leftarrow PITentry\ seq$
 - 19: Send $InterestNACK$ downstream via $\forall i \in PITentry\ in\ faces$
 - 20: $PITentry\ nonces \leftarrow Interest\ nonce$
 - 21: $PITentry\ lifetime \leftarrow Interest\ lifetime$
 - 22: $PITentry\ seq \leftarrow Interest\ seq$
 - 23: Send $Interest$ upstream
 - 24: **end if**
-

```

24: if InterestNACK was sent downstream at r then
25:   if InterestNACK arrived at consumer i then
26:     Consumer i send successive Interests using LFBR
27:   end if
28:   if Data arrived at r then
29:     Data advertisereplica  $\leftarrow$  1
30:     if Data arrived at consumer j then
31:       Consumer j advertise replica using NLSR
32:     end if
33:   end if
34: end if

```

4.2.2 Interest and Data Packets

To enable the communication between consumers within the same ISP/AS network and the satisfaction of successive Interests, an Interest NACK packet will be sent downstream to the consumers whose Interests have been aggregated at a CS-ERA border router, called *subordinate consumers*. An Interest NACK packet is used to inform subordinate consumers to suspend their operation of propagating Interests beyond the boundaries of the local network, and to bound their successive Interests to be satisfied within the ISP/AS network using the Locality-First Best Route (LFBR) forwarding strategy (Section 4.2.7). To enable the communication between consumers whose Interests have not been aggregated and a CS-ERA border router, no modifications to the format of an Interest are necessary. Nonetheless, a Data packet must be modified on either type of communication to include information about whether a consumer should act as a

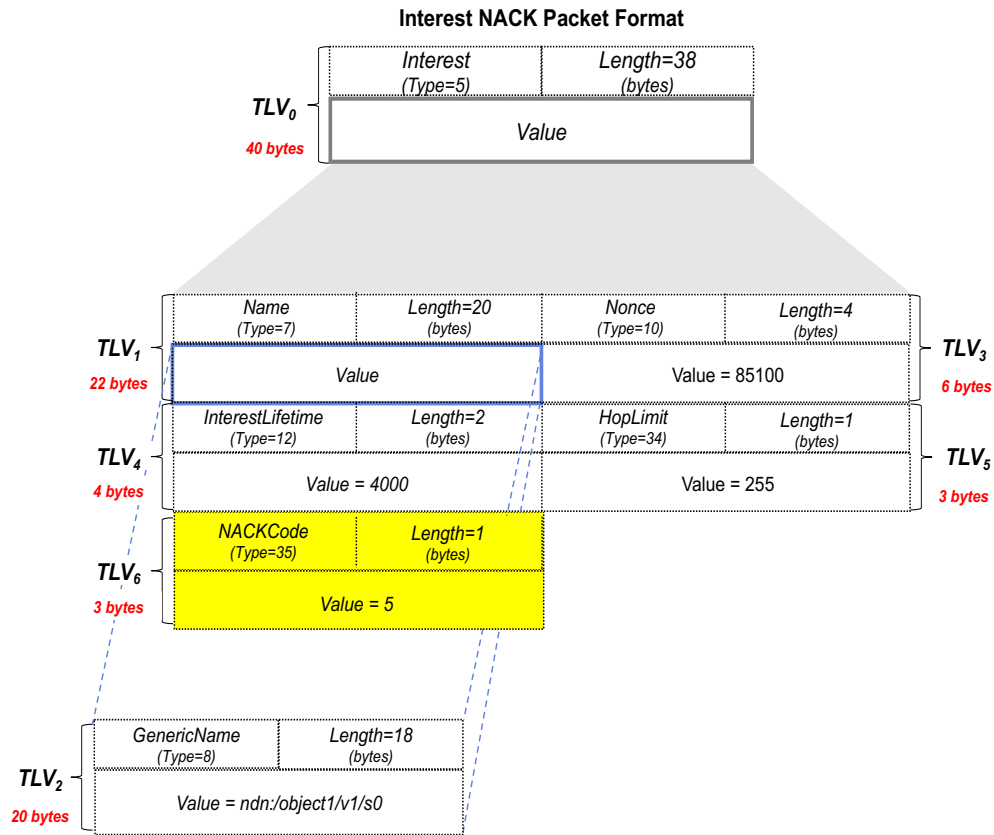


Fig. 4.2: Representation of the fields of an Interest NACK packet necessary and/or related to the context of this thesis for the operation of CS-ERA at a border router.

temporary content source, also called *principal consumer*. Besides the modifications described below, both packet formats are equivalent to the packet formats introduced in Section 2.1.2. For this reason, the focus of this section lies on highlighting the differences between the packet formats. For ease of understanding, Fig. 4.2 and 4.3 provide a representation of an Interest NACK and a Data packet format.

- **Interest NACK:** An Interest NACK packet is a form of Interest packet that NDN routers may propagate downstream to consumers, if they are unable to propagate

the Interest upstream to a potential content source(s) [Yi a]. As an example, an Interest NACK packet may be propagated downstream if an NDN router maintains no information in its FIB about how to propagate an Interest upstream. An Interest NACK packet consists of the same name and nonce as the Interest received, and for which the Interest NACK is sent. Hence, an Interest NACK packet is assumed to satisfy all entries within a PIT table of an NDN router that have been aggregated under the name for which the Interest NACK was sent. An Interest NACK packet contains a *nackcode* field used to indicate the reason why the Interest could not be propagated upstream. This information is used by the downstream routers and consumers to determine their future action. Similar to the Content Type field of a Data packet, a non-negative integer within the range $\{0, 2^8 - 1\}$ should be sufficient to support the current and future nackcodes. Hence, the size of this field could be considered equal to 1 byte. To support the communication between consumers within the local ISP/AS network, a nackcode equal to 5 has been defined as part of the error numbers that Interest NACK packets contain. This will allow an Interest NACK packet to be propagated through the network until reaching the corresponding consumer(s).

- **Data:** To support the communication between a CS-ERA border router and a consumer, additional information needs to be added to a Data packet to inform a principal consumer(s) to temporarily act as a content source(s), or a cache, for the retrieved content. The concept of consumers acting as temporary caches has been previously proposed in the context of in-network caching [Ahlgren b, Zhu], and is no different from the assumption of NDN routers acting as temporary content sources. To this end, and since the principal consumer does not act as a new

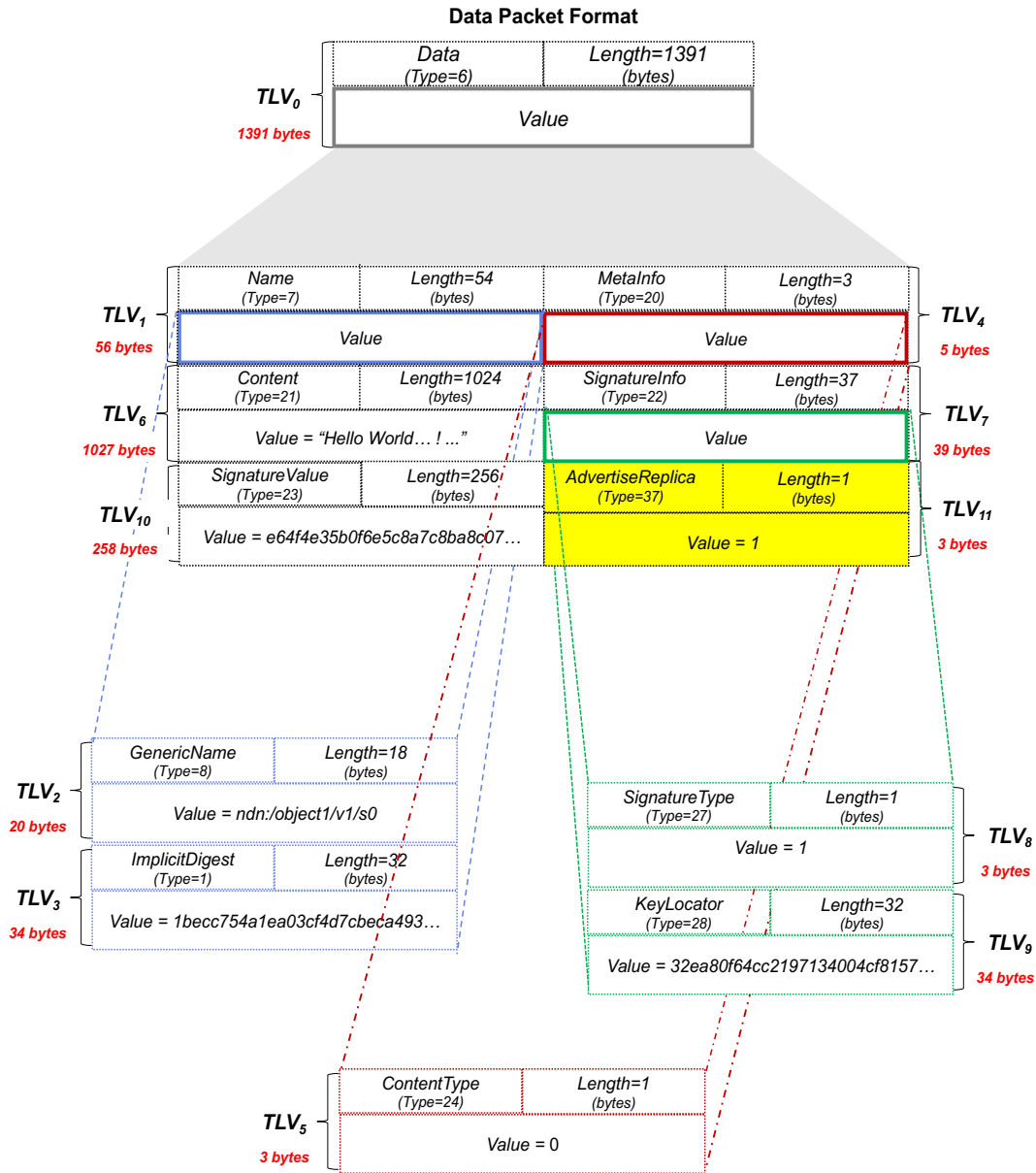


Fig. 4.3: Representation of the fields of a Data packet necessary and/or related to the context of this thesis for the operation of CS-ERA at a border router.

content producer, no additional security mechanisms are necessary to ensure that the content provided by a principal consumer is trustworthy [Zhang a]. Similar to the `nackcode` field of an Interest NACK, a hard-coded identification number may be used to inform the principal consumer(s) to temporarily act as a cache, by including an additional field in the Data packet format, called *advertisereplica*. A non-negative integer within the range $\{0, 1\}$ should be sufficient to support the status of this field, e.g. a value of 0 may mean that the consumer does not need to advertise its content within the local ISP/AS network, while a value of 1 may mean that the consumer is expected to act as a principal consumer. Hence, the size of this field could be considered equal to 1 byte. By enabling principal consumers to act as temporary caches, CS-ERA increases the availability of content within the local ISP/AS network, while eliminating the inter-network traffic.

4.2.3 PIT Entry Format

To support the functionality of CS-ERA, the format of a PIT entry at a CS-ERA border router has been modified to consist of the following information. This information is mainly extracted from the Interest packet format upon the arrival of an Interest at a CS-ERA border router. Besides these modifications, no modifications to the entries of a PIT at the edge and intermediate routers within an ISP/AS network are necessary.

- *prefix*: A content identifier used to identify an object resource, for which an Interest has been received. This information is extracted from the name field of an Interest. A prefix should not include the sequence number `seq` of a chunk of an object resource. If that number exists, it should be discarded.
- *seq*: A sequence number used to identify a chunk of an object resource, attached

to the prefix name component for which an Interest has been received. A common practise in NDN is to create human-readable names using the UTF-8 encoding scheme. Following this convention, a sequence number is represented using a non-negative numerical identifier. Similar to the prefix, this information is extracted from the name field of an Interest.

- ***infaces***: A list of incoming faces through which an Interest(s) for this prefix and sequence number seq, that either generated or updated the entry, has been received since the creation of the entry.
- ***outfaces***: A list of outgoing faces, through which an Interest(s) for this prefix and sequence number seq has been propagated to a content source(s) since the creation, or the update, of the entry.
- ***lifetime***: A time counter that once expired, triggers the removal of the entry. A lifetime is initialised and updated upon the arrival of an Interest for this prefix, by extracting the value of the equivalent field from the Interest packet format.
- ***nonces***: A list of nonces of the Interests aggregated by the mechanism of CS-ERA for this prefix and sequence number seq since the creation, or the update, of the entry. Nonces are used to detect duplicated Interests that have been forwarded through multiple faces within a network and discard them. This information is extracted from the equivalent field of an Interest.

4.2.4 Upstream Process

The term upstream process refers to the arrival of Interests at an NDN router. Following the representation of the Interest and Data packet formats presented in Fig. 2.1, 4.2 and 4.3, Fig. 4.4 provides a representation of the upstream process at a CS-ERA border router for the 5th chunk of the object resource *ndn:/object1/v1*, described as follows: Upon the arrival of an Interest a router will perform a CS lookup (1). If a CS entry that matches the content identifier of the Interest exists, the content will be propagated to the consumer through the face at which the Interest arrived (2), i.e. *face₂*. If otherwise, a router will perform a PIT lookup (3). If a PIT entry that matches the prefix of the content identifier of the Interest does not exist, a PIT entry will be created (4). Once a PIT entry is created, the Interest will be propagated upstream to a content source(s) according to the information in the FIB (5), i.e. *face₄*. If a FIB entry that matches the content identifier of the Interest does not exist, the Interest will be discarded (6). If a PIT entry that matches the prefix of the content identifier of the Interest exists, the PIT entry will be updated depending on the sequence number *seq* of the Interest. If the sequence number *seq* of the Interest is equal to the sequence number *seq* of the PIT entry (7), the Interest will be aggregated. The existing PIT entry will be updated according to the information included in the incoming Interest, such as its lifetime and nonce (8). The incoming face on which the Interest has been received will be added in the list of infaces. If the sequence number *seq* of the Interest is lower than the sequence number *seq* of the PIT entry (9), the PIT entry will be updated (10). The incoming face on which the Interest has been received will not be added in the list of infaces. Yet, the lifetime of the PIT entry will be updated to indicate that more consumers are interested in receiving this object resource. An Interest NACK will be sent downstream through

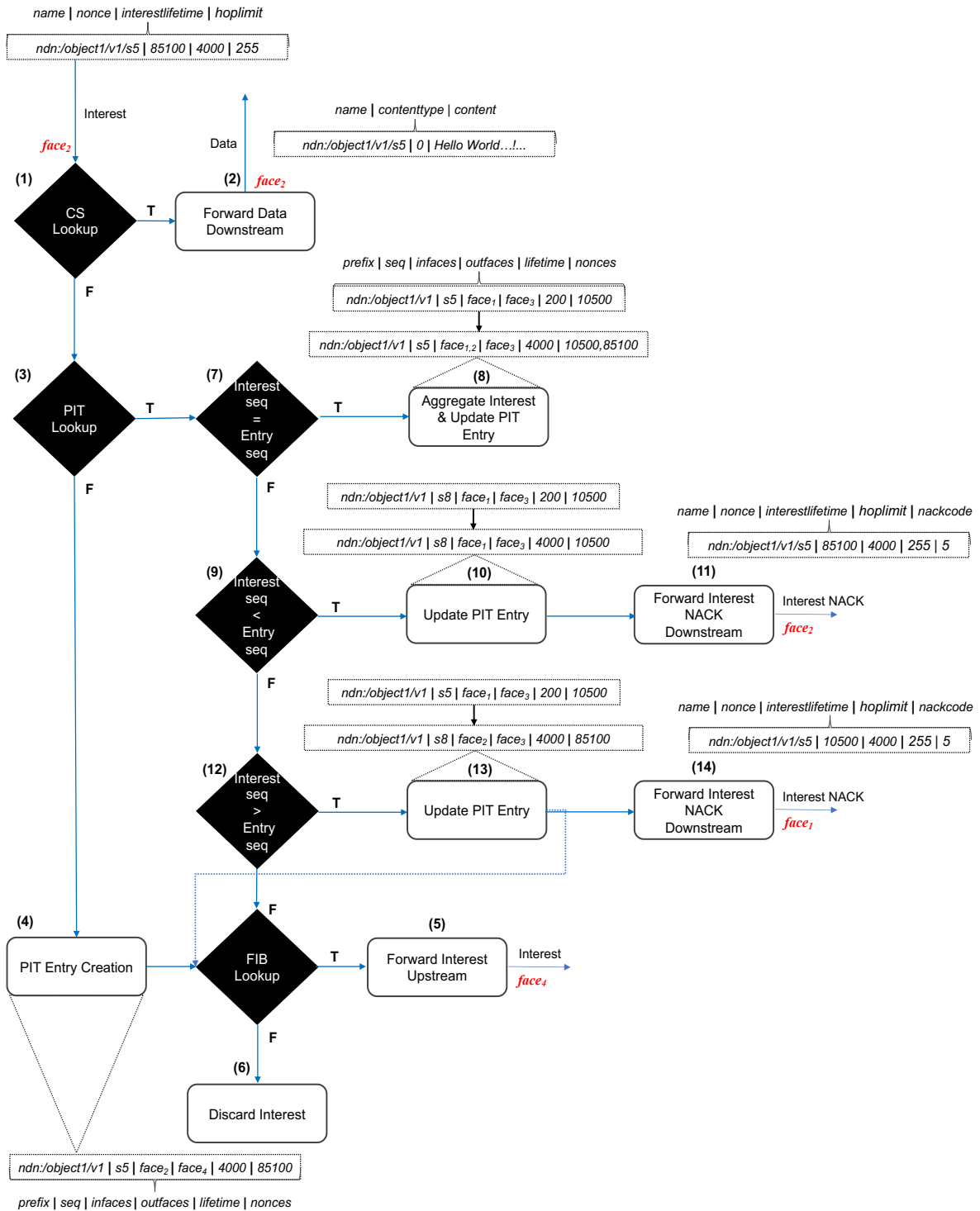


Fig. 4.4: Upstream forwarding process of an Interest at a CS-ERA border router.

the face at which the Interest arrived to inform the consumer(s) to bound its successive Interests locally using the LFBR forwarding strategy (11). If otherwise (12), the PIT entry will be updated as if a new PIT entry was created according to the information included in the Interest received (13). The lifetime of the PIT entry will be updated to indicate that more consumers are interested in receiving this object resource. Once the PIT entry is updated, the Interest will be propagated upstream to a content source(s) according to the information in the FIB (5) and (6). An Interest NACK packet will be sent downstream through the infaces recorded in the PIT entry before the update, to inform the consumer(s) whose Interest(s) has either created or updated the PIT entry to bound their successive Interests locally using the LFBR forwarding strategy (14).

4.2.5 Downstream Process

The term downstream process refers to the arrival of Data at an NDN router. Following the representation of the upstream process in Fig. 4.4, Fig. 4.5 provides a representation of the downstream process at a CS-ERA border router, described as follows: Upon the arrival of Data a router will perform a PIT lookup (1). If a PIT entry that matches the prefix and sequence number *seq* of the Interest does not exist, the packet is unsolicited and it will be discarded (2). If otherwise, a replica of the Data will be propagated to each of the consumers interested in retrieving this content through the faces in the infaces list (4), i.e. *face*₁ and *face*₂. To notify a consumer to act as a principal consumer within the local ISP/AS network, assuming that request aggregation has been applied and at least one Interest NACK packet has been sent downstream, a CS-ERA border router will update the advertise replica field of each replica from 0 to 1 before propagating the replicas downstream (3). After the propagation of Data, the PIT entry will be deleted.

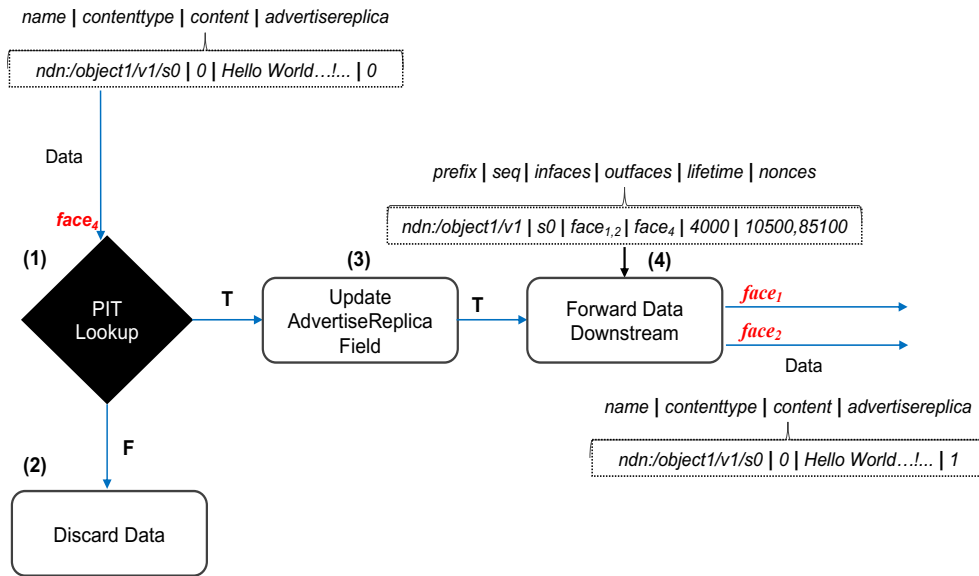


Fig. 4.5: Downstream forwarding process of Data at a CS-ERA border router.

4.2.6 Principal Consumer

A principal consumer is a consumer within the local ISP/AS network that has been notified to act as a temporary content source, or a cache for some content. Its selection is based on the maximum number of chunks that a consumer has retrieved, and whose Interests have been aggregated at a CS-ERA border router. This number is determined by comparing the sequence number *seq* of the Interests received at a CS-ERA border router to the sequence number *seq* of the corresponding PIT entry, following the upstream process defined in Section 4.2.4. Consequently, a principal consumer is responsible for retrieving the remaining chunks of an object resource that have not been retrieved yet by neither itself, nor the subordinate consumers. This way, CS-ERA ensures that the principal consumer(s) is able to satisfy the Interests issued within the local ISP/AS network for which the sequence number *seq* is lower than the maximum sequence number *seq* of

the chunks it possesses. By doing so, CS-ERA ensures that the number of Interests to be propagated beyond the boundaries of the local ISP/AS network is bound to a minimum.

Upon the reception of a Data packet used to inform a consumer to act as a principal consumer, a consumer will advertise the prefix of the cached content to its local NDN router using a *Prefix Link State Advertisement (LSA)*. Advertising the content will allow the principal consumer to share its content within the local ISP/AS network and to satisfy successive Interests of subordinate consumers. LSAs are propagated by the *Named-data Link State Routing (NLSR)* protocol, supported by the NDN architecture [Hoque]. In this sense, the operation of a principal consumer is no different to the operation of a producer. Nonetheless, a principal consumer does not act as a new content producer, but as a temporary source of content that hosts a replica. The concept of consumers acting as temporary caches has been previously proposed in the context of in-network caching [Ahlgren b, Zhu], and is no different from the assumption of NDN routers acting as temporary content sources. Hence, no additional security mechanisms are necessary to ensure that the content provided by a principal consumer is trustworthy [Zhang a]. Upon the reception of a Prefix LSA, or an Adjacency LSA used to advertise connectivity information of the routers within a network, an NDN router will verify whether the LSA is originated from an authorised entity within the local ISP/AS network. For this purpose, NLSR requires the existence of a hierarchical trust model [Yu b], rooted at the *trust anchor*, i.e. the network administrator. A trust model defines relationship rules between names and public-private key pairs used by producers and consumers to sign and verify content. A trust model is used to verify the authenticity of the key used in the signing process of the LSA. If the verification is successful, an NDN router will propagate the LSA further within the ISP/AS network.

To enforce fast convergence of the FIB tables, NLSR ensures that each Prefix LSA

contains only one name prefix, instead of all name-prefixes that an NDN router may maintain. In addition to this, and to prevent the use of invalid entries registered within the FIB of NDN routers, each Prefix LSA contains an *isValid* field that states whether a prefix is reachable. In the event that a principal consumer is unwilling to continue acting as a temporary content source, a de-register Prefix LSA may be advertised to its local NDN router for the given prefix, by setting its *isValid* field equal to 0. Upon the reception of a Prefix LSA, the local NDN router will verify whether the LSA is originated from an authorised entity within the local ISP/AS network by consulting the hierarchical trust model in place. If the verification is successful, the NDN router will propagate the LSA further within the network. Upon the reception of a valid de-register Prefix LSA, an NDN router will remove the corresponding entry from its FIB. Besides the use of de-register Prefix LSAs, a lifetime is used to ensure that a Prefix LSA will eventually become invalid. To ease the calculation of lifetimes, a timer may be used complementary to the *isValid* field of a Prefix LSA. This timer is expected to be initialised by a principal consumer, to indicate the time for which it wishes to act as a temporary content source. To prevent unnecessary overheads introduced due to the propagation of LSAs within the local ISP/AS network and the convergence of FIB tables, a local NDN router may reject a Prefix LSA received by a principal consumer if the lifetime contained is lower than a threshold defined a priori. To faster recover from cases where a prefix becomes unreachable, a type of Interest packet called *info*, used by NLSR to detect failures on the links or the routing process, may be sent from a local NDN router to a principal consumer periodically. If the Interest times out the local NDN router may try again a few times in short time intervals and continue sending Interests in longer time intervals.

4.2.7 FIB and Forwarding Strategy

To distinguish between the faces registered in the FIB of an NDN router to temporary replicas of content located within the boundaries of the local ISP/AS network, and permanent content sources located beyond the local network to the Internet infrastructure, FIB entries may further be enriched by a *locality* attribute. This is similar to the attribute used in a FIB to support the *Colouring Scheme (Green/Yellow/Red)* forwarding strategy [Yi a]. To support the Colouring Scheme forwarding strategy, FIB entries maintain information derived from both the NLSR routing protocol and the forwarding strategy to support adaptive forwarding decisions by identifying the current forwarding status of a face. As an example, a Green status means that a face can deliver Data back to consumers, a Yellow status means that it is unknown whether a face can deliver Data back to consumers, while a Red status means that a face can not deliver any Data.

To support the functionality of CS-ERA and the satisfaction of successive Interests within the local ISP/AS network by a principal consumer(s), an extension of the Best Route (BR) forwarding strategy (Section 2.3.5), called *Locality-First Best Route (LFBR)* forwarding strategy, has been adopted. According to the functionality of this forwarding strategy, the BR forwarding strategy is used to propagate an Interest to a temporary replica within the local ISP/AS network, if one exists. For this purpose, only the faces of a FIB entry whose locality attribute is *local* are used. If otherwise, the BR forwarding strategy is used to propagate an Interest to a content source beyond the local network. For this purpose, only the faces of a FIB entry whose locality attribute is *global* are used.

The LFBR forwarding strategy ensures that an Interest will reach either a temporary cache that possesses a replica of the content or a content source. This in turn ensures that consumers are able to retrieve content and recover from cases where the FIB entries

of NDN routers have not been populated yet, or they are invalid. Similar to the use of an Interest NACK packet as described in Section 4.2.2, to inform consumers to suspend their operation of propagating Interests to producers beyond the boundaries of the local ISP/AS network, an Interest NACK packet may be used to inform consumers that all faces whose locality attribute is *local* have been exhausted and no Data has been retrieved. This operation is already supported by Interest NACK packets, when an NDN router maintains no information in its FIB about how to propagate an Interest. Hence, no additional hard-coded identifiers need to be defined. Upon the reception of an Interest NACK packet that indicates that all local faces have been exhausted, consumers have no other option but to propagate their Interests beyond the local network.

4.3 System Model

In this section, a description of the system model used in the evaluation of CS-ERA using the ndnSIM(v1) simulator, against the original request aggregation mechanism, is given. This system model is similar to the system model used in the quantitative comparison of caching policies in Section 2.3.7, including the network topologies (Section 2.3.3), applications (Section 2.3.4) and architectural components (Section 2.3.5). Hence, to avoid repetition, the focus of this section lies on highlighting the differences between the two system models. A summary of the system model used is presented in Table 4.1.

4.3.1 Applications

Referring to Section 2.3.4.1, a consumer application has been defined to request objects based on a Weibull popularity distribution of *shape=0.513* and *scale=6010* [Cheng , Pentikousis], and to issue object requests according to an exponential distribution of

$mean = 1.0$ and $upper_{limit} = 2.0$ [Katsaros ,Cooper]. The number of Interests expressed for an object resource is determined by the size of the object resource divided by the chunk size. The size of an object resource is assumed to follow a normal distribution with a mean of 10MB and a standard deviation of 9.8MB [Gill ,Zhou a]. The size of a chunk is defined to be equal to 10KB [Muscarello b,Carofiglio a]. Hence, an object resource is expected to be equal to 10^3 successive Interests on average. Each successive Interest is scheduled within 0 seconds from the time the last Data was received. The aforementioned parameters have been adopted in previous studies and suggested by the ICN research community for the evaluation of UGC traffic.

This setup has been shown to introduce a significant computational and processing overhead per simulation run [Project b,Shi a] with regard to the evaluation of caching policies presented in Section 2.3.7. Besides the time and processing limitations encountered in the evaluation of full object resources, a memory consumption problem has also been encountered due to the number of CS entries to be maintained per simulation run [Shi a]. Due to these limitations, the size of an object resource has been deliberately set to be equal to a single Interest [Xie ,Psaras a,Xiaoyan]. The adaptation of single-Interest object resources has been based, among others, on the fact that the performance of the caching policies in question would not be affected since neither evaluation metric is dependent to the number of Interests issued per object resource. Even though this assumption holds true for the evaluation of caching policies, it does not hold true for the evaluation of CS-ERA against the original request aggregation mechanism. In this evaluation, the researcher is interested in measuring the number of Interests of an object resource that have been satisfied within the ISP/AS network. This measurement is used to define the inter-network traffic reduction in terms of Interests. For the operation of the producer application no modifications are necessary, as defined in Section 2.3.

4.3.2 Architectural Components

The NDN architecture consists of the following architectural components: a CS, a PIT and a FIB. The ability of an NDN router to cache content on delivery paths, supported by the CS architectural component, is expected to cancel the benefit of a request aggregation mechanism since on-path caching allows Interests to be satisfied locally at a router and prevents them from being propagated upstream. For this reason, and similar to a number of studies conducted in the area of request aggregation [Panwar , Shannigrahi], the operation of a CS has been disabled to avoid any complications drawn from the impact of a caching policy on the performance of a request aggregation mechanism and to focus on the comparison of the mechanisms in question. Moreover, the co-existence of a caching policy and a request aggregation mechanism at an NDN router is not necessary; either of these mechanisms may be disabled or not supported. Nonetheless, to be able to verify and to measure the impact of on-path caching on the performance of the mechanisms in question, the default caching policy of the NDN architecture, i.e. LCE (Section 2.2.3.1) is used to enable caching within an NDN network. All routers within a network perform caching on a chunk naming granularity, as defined by the NDN architecture.

The LCE caching policy is expected to act as a benchmark to the evaluation of future caching policies in the context of the feature interaction of on-path caching and request aggregation. At this point it is worth noting, that the higher the benefit of an on-path caching policy, the lower the benefit of a request aggregation mechanism [Dabirmoghaddam]. Hence, caching policies such as PC+ or PLbC, (sections 2.2.3.2 and 3.2.2) are expected to result in a lower benefit for either mechanism, compared to LCE. Due to the memory limitations encountered in the evaluation of full object resources, the number of CS entries to be maintained per simulation run has been reduced to the maximum

Network Parameters	
Parameter	Symbol Value
# of routers in network	$ N $ 75, 97 [Badov ,Bernardini]
# of intermediate routers in network	$ B $ 19, 39 [Afanasyev b]
# of edge routers in network	$ G $ 56, 58 [Afanasyev b]
# of border routers in network	$ B $ 1
# of consumers at edge router i	u_i $\forall u_i, i \in G \sim U(100, 200)$ [Detti b, Li e, Saha]
# of producers in network	p 1 [Arianfar ,Psaras a]
Link capacity/bandwidth	BW 40GB [Psaras a]
Link delay	d 10ms [Afanasyev d]
Cache size in chunks/Data	cs $cs \in \{10^2, 10^3, 10^4\}$ [Arianfar , Lee , Perino]
# distance to producer in hop-counts	hc $hc \in \{1, 3, 5\}$ [Begtasevic ,Fei ,Kasiviswanathan]
Traffic Model Parameters	
Parameter	Symbol Value
Catalog size	$ O $ $10^6, 10^8$ [Pentikousis]
Object size in KB	o_i $\forall o_i, i \in O \sim N(10^4, 9.8 \times 10^3)KB$ [Gill ,Zhou a]
Object size in chunks/Data	C 10^3 [Xie ,Psaras a,Xiaoyan]
Chunk size in KB	ch $10KB$ [Muscariello b,Carofiglio a]
Object Popularity Parameters	
Parameter	Symbol Value
Shape of Weibull distribution	$shape$ 0.513 [Cheng ,Pentikousis]
Scale of Weibull distribution	$scale$ 6010 [Cheng ,Pentikousis]
Request Arrival Parameters	
Parameter	Symbol Value
Mean of Exponential distribution	$mean$ 1.0 [Katsaros ,Cooper]
Upper limit of Exponential distribution	$upper$ 3.0 [Katsaros ,Cooper]

Table 4.1: Summary of the system model and the evaluation parameters used in the evaluation of the PC+, HPC, CAC, LCE and LCL caching policies.

possible size, i.e. equal to 10^5 chunks. According to the guidelines suggested by the ICN research community, a cache size of 10^5 chunks is 10 times lower than the maximum size of caches suggested, i.e. equal to 10GB [Arianfar , Lee , Perino].

The PIT architectural component has been modified to follow the design of a PIT entry at a CS-ERA border router, as described in Section 4.2.3. In addition to the PIT architectural component, a number of modifications have been applied to the FIB architectural component and the BR forwarding strategy, as described in Section 4.2.7. Even though these modifications are part of the proposed design, they are based on the functionality of the NLSR routing protocol, a real-world application available in ndnSIM(v2). ndnSIM(v2) is different to ndnSIM(v1) in a number of ways [Mastorakis b]. Hence, a comparison between the results obtained from ndnSIM(v1) and ndnSIM(v2) would be invalid. Moreover, ndnSIM(v2) has been shown to increase both the processing and memory requirements of ndnSIM(v1). As an example, the processing overhead and memory consumption for a CS entry in ndnSIM(v1) is equal to 12.84 seconds and 0.75KB at minimum, while the equivalent values in ndnSIM(v2) are equal to 23.56 seconds and 5.40KB. Due to these reasons, an examination of the impact of the NLSR protocol on the performance of a network has been excluded from this evaluation. A discussion that highlights the complications to be expected from its operation is presented in Section 4.5.4.

4.4 Evaluation

In this section, a description of the evaluation metrics used in the evaluation of CS-ERA against the original request aggregation mechanism is given. This description is followed by a presentation on the evaluation results. The focus of the evaluation lies on measuring the probability of request aggregation occurring within a local ISP/AS network and the

inter-network traffic propagated beyond the boundaries of this network to the Internet infrastructure. The evaluation aims to quantify the ability of either mechanism to satisfy the objectives of this thesis: The reduction of the network traffic within a local ISP/AS network and the traffic propagated beyond the boundaries of this network to the Internet infrastructure. To examine the effect of the location of a content source indicated by its distance from the ISP/AS network measured in hop-counts hc , a range of hop-count values have been considered, i.e. $hc \in \{1, 3, 5\}$, represented by introducing an additional delay at the CS-ERA border router in the simulator. The selection of these values has been based on a number of measurement studies on the hop-count metric of the Internet infrastructure, while taking into account the impact of existing technologies such as CDNs and edge-computing [Begtasevic ,Fei ,Kasiviswanathan].

For ease of reference, the terms *ORIGINAL* and *CS-ERA* are used to refer to the original request aggregation mechanism and the mechanism proposed in this chapter, respectively. The term *approach* is used to refer to either mechanism and evaluation system. For ease of comparison, the mean and standard deviation (SD) values of the evaluation metrics of an approach are used, derived from 10 simulation runs with regard to the parameters presented in Table 4.1. The SD values are presented on top of the mean values of an evaluation metric. This form of representation of the mean and SD values may exceed the maximum value of an evaluation metric. The SD values are used as a complementary descriptive measurement to the mean values to identify the variance on the behaviour of the mechanisms and the effect of the evaluation parameters on their performance [Hassani].

4.4.1 Evaluation Metrics

CS-ERA has been evaluated against ORIGINAL using the following evaluation metrics, calculated at the consumer applications within a network:

- **Aggregation Frequency (AF):** The *aggregation frequency* metric refers to the number of object resources aggregated by a request aggregation mechanism, normalised by the number of object resources issued within a network. This metric indicates the tendency of a request aggregation mechanism to aggregate object resources or the probability of request aggregation occurring within an NDN network. This metric is complementary to the aggregation-efficiency ratio metric.
- **Aggregation-Efficiency Ratio (AER):** The *aggregation-efficiency ratio* metric refers to the number of Interests of an object resource aggregated and satisfied within the local ISP/AS network, normalised by the number of Interests issued per object resource, whether these have been satisfied or not. This metric indicates the offload of a producer and the reduction rate of inter-network traffic. Preferable request aggregation mechanisms should result in an increase of this metric.

4.4.2 Evaluation Results

Fig. 4.6 and 4.7 illustrate the mean AF and SD values of an approach with regard to the catalog size $|O|$ and the distance of a content source from the ISP/AS network in hop-counts hc for the AS-3257 network topology and the AS-3967 network topology, respectively, when caching is disabled, and $|O| \in \{10^6, 10^8\}$ and $hc \in \{1, 3, 5\}$. CS-ERA outperforms ORIGINAL by 11-15%, depending on the network topology and the parameters used. The mean AF of ORIGINAL is equal to 14% when $hc \in \{1, 3\}$, and equal to 13% when $hc = 5$ for the AS-3257 network topology. The mean AF of ORIGINAL

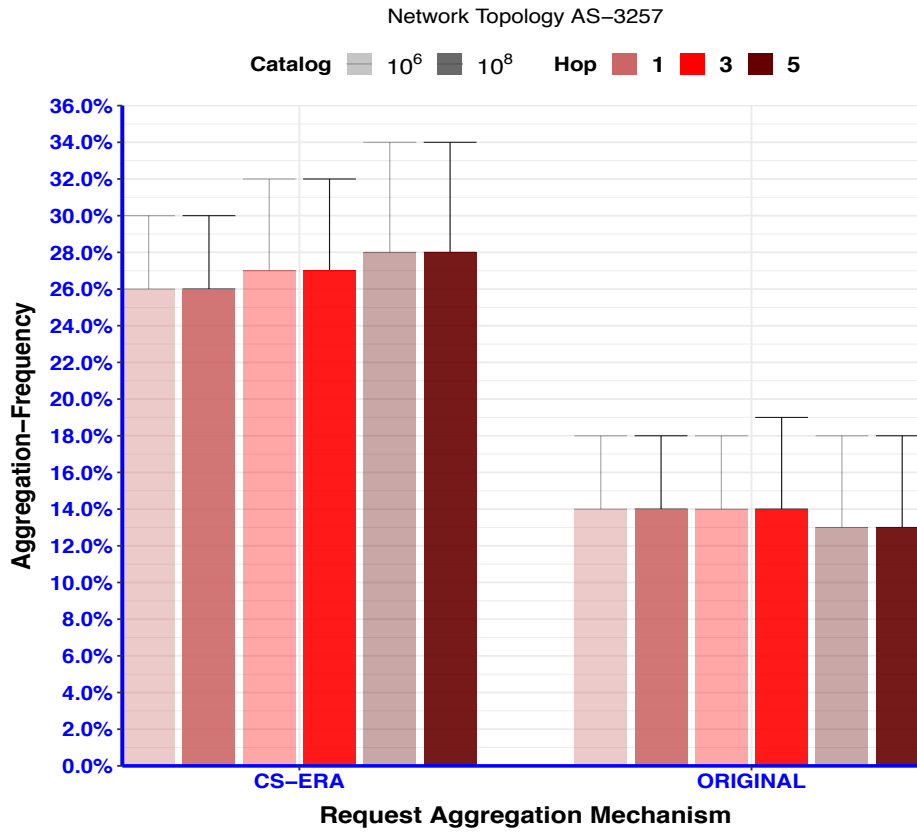


Fig. 4.6: Representation of the mean AF and SD values of an approach for the AS-3257 network topology, when caching is disabled and $|O| = \{10^6, 10^8\}$ and $hcc \in \{1, 3, 4\}$.

for the AS-3967 network topology is 1% higher than the equivalent values concluded for the AS-3257 network topology. The mean AF of CS-ERA is equal to 26%, 27% and 28% when $hcc \in \{1, 3, 5\}$ for either network topology. The SD values of both CS-ERA and ORIGINAL range between 4-6% for both network topologies, while a higher variance is concluded as the distance of a content source from the ISP/AS network increases.

According to both figures, a number of conclusions may be withdrawn as follows:

- The AF metric of ORIGINAL decreases as the number of routers on the forwarding and the delivery paths increases. A higher number of routers to which request

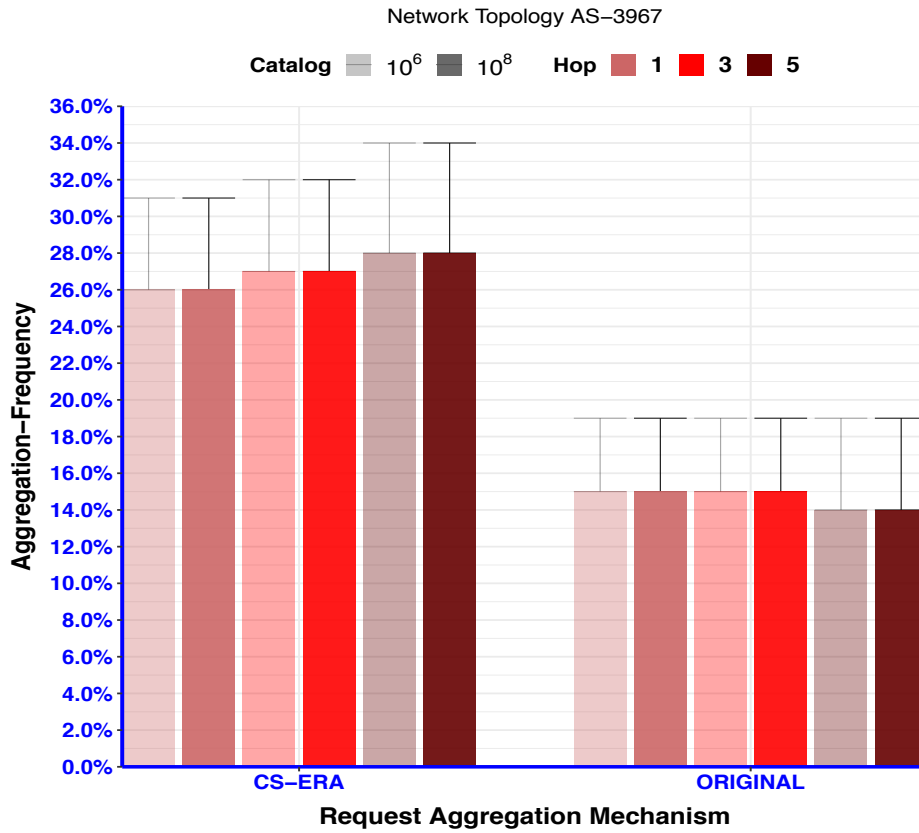


Fig. 4.7: Representation of the mean AF and SD values of an approach for the AS-3967 network topology, when caching is disabled and $|O| = \{10^6, 10^8\}$ and $h \in \{1, 3, 4\}$.

aggregation may be applied increases the probability of request aggregation occurring within an NDN network. This claim may be verified by consulting the mean AF metric of ORIGINAL for both network topologies.

- An increase in the number of routers on the forwarding and the delivery paths does not affect the AF metric of CS-ERA, since CS-ERA is applied to the border routers of an ISP/AS network.
- The AF metric and SD values of CS-ERA increase, while the equivalent values of

ORIGINAL decrease as the distance of a content source from the ISP/AS network increases. This outcome contradicts the theory that an increase in the lifetime of PIT entries is expected to increase the probability of request aggregation occurring within an NDN network. This contradiction may be explained as follows: Since CS-ERA performs request aggregation on an object naming granularity rather than a chunk naming granularity, a higher increase in the probability of request aggregation occurring within an NDN network is concluded compared to ORIGINAL.

- Since CS-ERA performs request aggregation on an object naming granularity rather than a chunk naming granularity, CS-ERA concludes to higher AF values faster than ORIGINAL. This claim may be verified against Fig. 4.8 and 4.9, which illustrate the mean number of object requests issued by a consumer in simulations.
- The AF metric of either mechanism is independent of the catalog size, since the difference between the catalog sizes considered is insignificant to affect this metric. This outcome may be explained similar to Section 2.3.7.

Fig. 4.10 and 4.11 illustrate the mean AER and SD values of an approach with regard to the catalog size $|O|$ and the distance of a content source from the ISP/AS network in hop-counts hc for the AS-3257 network topology and the AS-3967 network topology, respectively, when caching is disabled, and $|O| \in \{10^6, 10^8\}$ and $hc \in \{1, 3, 5\}$. CS-ERA outperforms ORIGINAL by 52-68%, depending on the network topology and the parameters used. The mean AER of ORIGINAL is equal to 38%, 44% and 48% when $hc \in \{1, 3, 5\}$ for the AS-3257 network topology. The mean AER of ORIGINAL is equal to 32%, 40% and 47% for the same hop-counts, respectively, for the AS-3967 network topology. The mean AER of CS-ERA is equal to 100% for either network topology, meaning that an object request aggregated by CS-ERA will be satisfied within the local

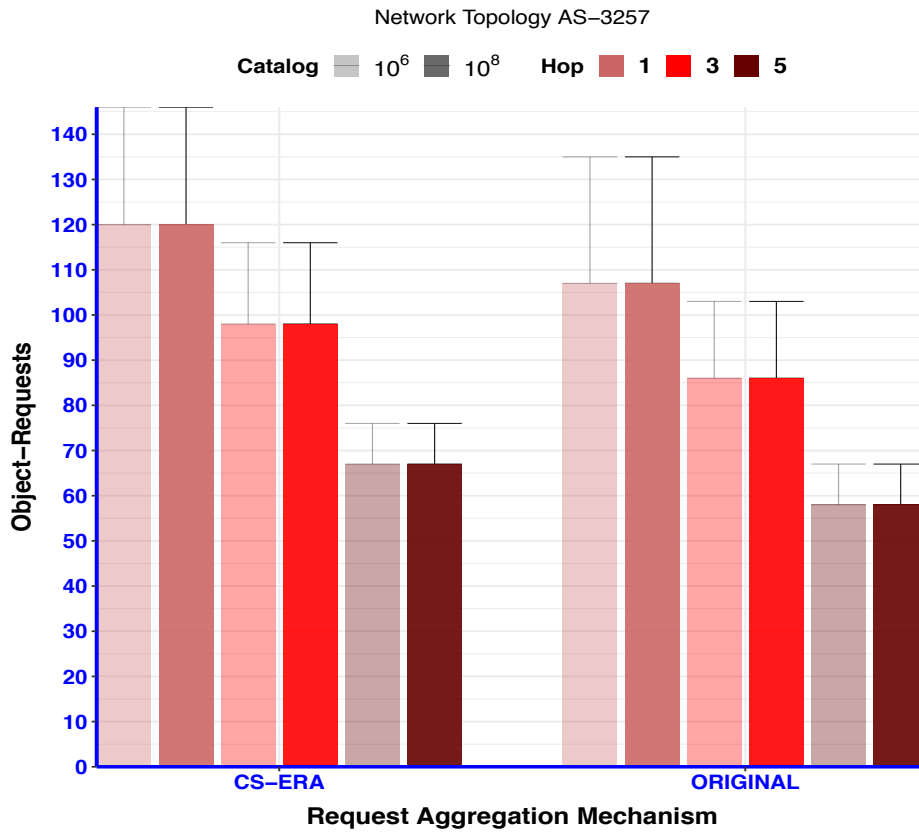


Fig. 4.8: Representation of the mean object requests issued by a consumer and SD values for the AS-3257 network topology, when caching is disabled and $|O| = \{10^6, 10^8\}$ and $hc \in \{1, 3, 4\}$.

ISP/AS network by a principal consumer(s). The SD values of ORIGINAL are between 28-35%, following the pattern of the mean values. The SD values of CS-ERA are equal to 0%, since no variation on its mean AER values is shown. Similar to the cache-based metrics, both the AF and AER metrics have been rounded to two decimal places, hiding any negligible differences between the values. Yet, an increase of the order of 10^{-4} on the mean AER values of CS-ERA is shown as the hc parameter increases.

According to both figures, a number of conclusions may be withdrawn as follows:

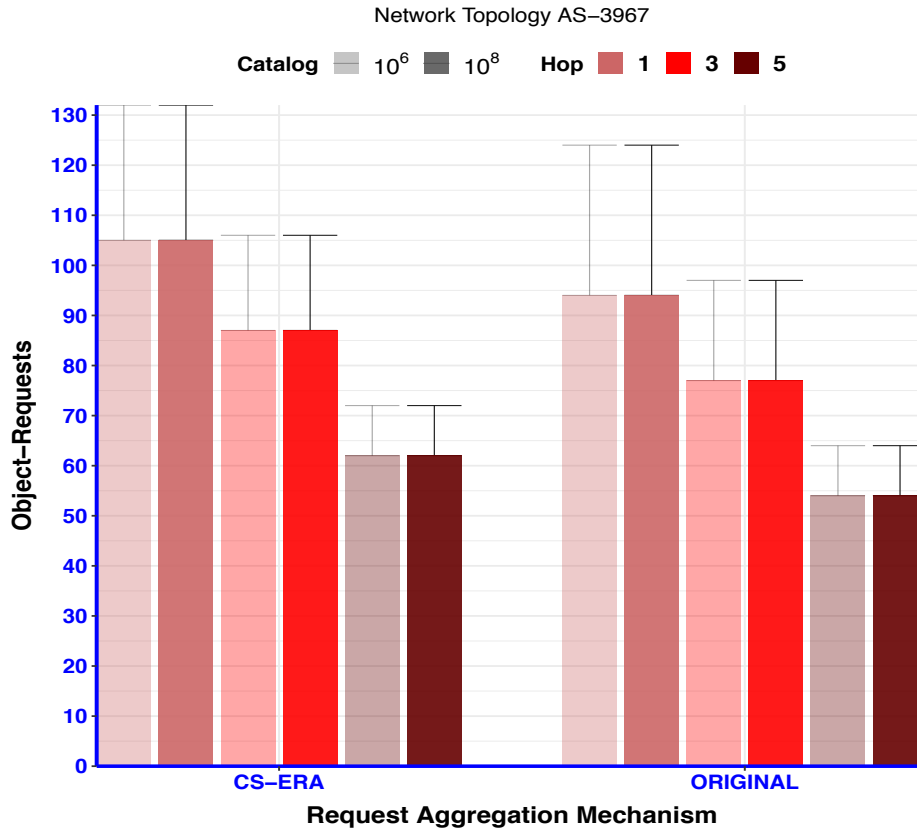


Fig. 4.9: Representation of the mean object requests issued by a consumer and SD values for the AS-3967 network topology, when caching is disabled and $|O| = \{10^6, 10^8\}$ and $h \in \{1, 3, 4\}$.

- The AER metric of ORIGINAL decreases as the number of routers on the forwarding and the delivery paths increases. This claim may be verified by consulting the mean AER and SD values of ORIGINAL for the AS-3257 network topology, against the AS-3967 network topology. Although this outcome is contradictory to the impact of the same parameter on the AF metric, it may be explained as follows: A higher number of routers increases the probability of request aggregation occurring, yet it also increases the occurrence of *single request aggregations*. This term

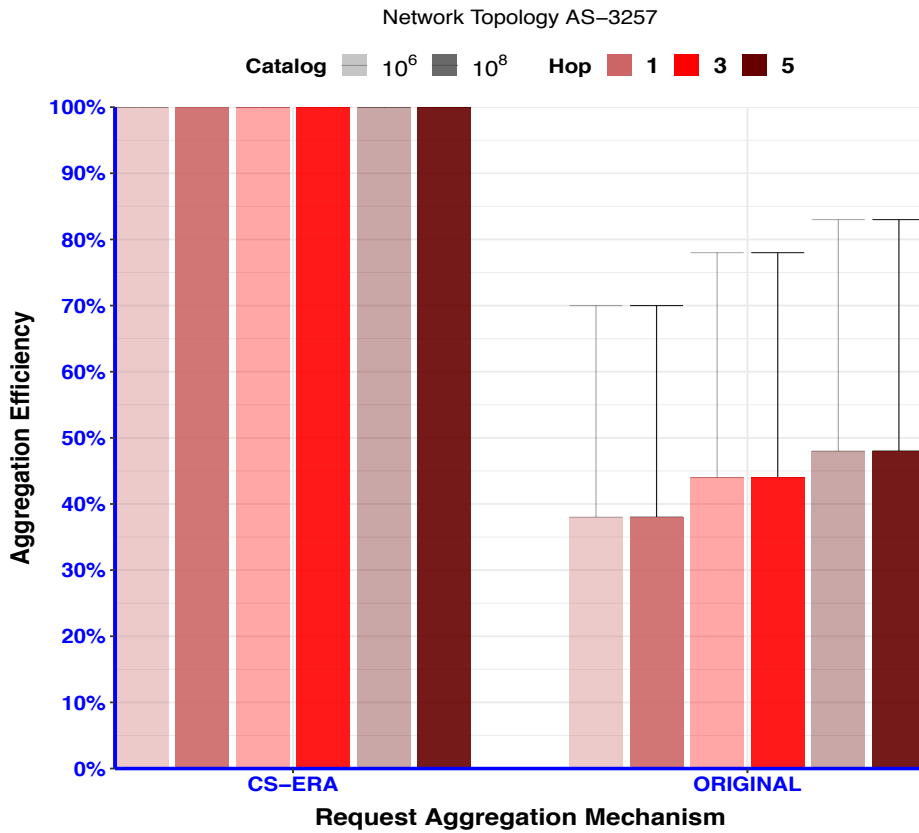


Fig. 4.10: Representation of the mean AER and SD values of an approach for the AS-3257 network topology, when caching is disabled and $|O| = \{10^6, 10^8\}$ and $hcc \in \{1, 3, 4\}$.

is used to refer to request aggregations of Interests that may occur at some point during the retrieval of an object resource, yet they will not occur again.

- The AER metric of CS-ERA is independent of an increase in the distance of a content source from the ISP/AS network.
- The AER metric and SD values of ORIGINAL increase as the distance of a content source from the ISP/AS network increases. This outcome may be explained similar to the AF values of CS-ERA: For each hop that a content source is placed away

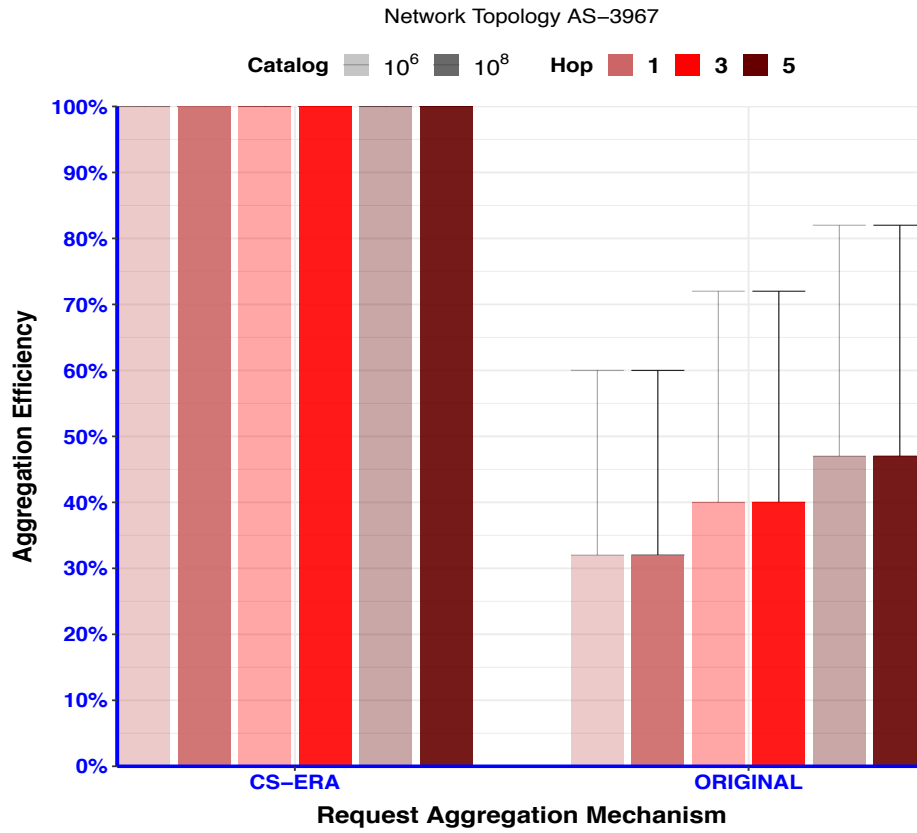


Fig. 4.11: Representation of the mean AER and SD values of an approach for the AS-3967 network topology, when caching is disabled and $|O| = \{10^6, 10^8\}$ and $hcc \in \{1, 3, 4\}$.

from the ISP/AS network, the time that an entry remains alive in the PIT of the intermediate routers increases proportionally. An increase in the lifetime of PIT entries is expected to increase the probability of request aggregation occurring within an NDN network and the AER values, increasing the variance of this metric.

- Similar to the impact of the catalog size on the AF metric of either ORIGINAL or CS-ERA, an equivalent impact is observed for the AER metric.

Fig. 4.12 illustrates the occurrence of single request aggregations within an ISP/AS

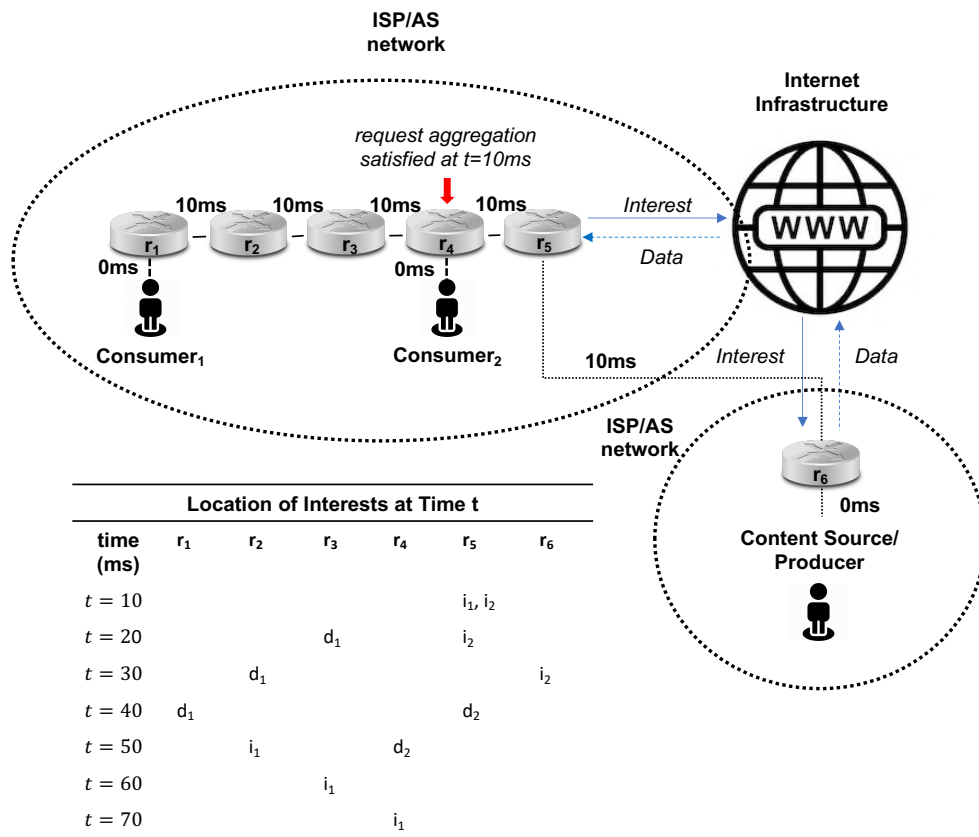


Fig. 4.12: Abstract representation of the functionality of ORIGINAL within an ISP/AS network when single request aggregations occur, illustrated by the co-existence of two Interests i_1 and i_2 at the same time.

network utilising ORIGINAL. Interests are able to reach the content source on a forwarding path of 5 routers. The content source is located in a different network, one hop away from the local ISP/AS network. Each of the links traversed between two routers introduces a propagation delay of 10ms, modifying this way the location of Interests and Data over time. The links between a router and a producer, or a consumer, called *local links*, do not introduce any delay. This is a design convention of the ndnSIM(v1) simulator used in the experiments [Afanasyev a]. To visualise the occurrence of request

aggregations, a table that illustrates the location of Interests i_x and Data d_x over time is used. The example starts at time $t = 10ms$ at router r_4 , assuming that two Interests, i.e. i_1 and i_2 , originated from $Consumer_1$ and $Consumer_2$, have been previously aggregated and that a Data packet has arrived at router r_4 to satisfy the Interests.

According to the table, a Data packet will be propagated at time=10ms from router r_4 to $Consumer_1$ and $Consumer_2$, called d_1 and d_2 , respectively. Since $Consumer_2$ is locally connected to router r_4 , no additional propagation delay will be introduced, while a successive Interest will be issued. This new Interest will reach the content source at time $t = 30ms$, where a new Data packet will be issued, that will reach router r_4 at time $t = 50ms$. Hence, by the time that $t = 50ms$, $Consumer_2$ will have issued its new Interest and received the corresponding Data. In contrast to d_2 , that was immediately delivered to $Consumer_2$, d_1 needs to traverse 3 intermediary routers to reach $Consumer_1$, introducing a delay of 30ms. Consequently, $Consumer_1$ will issue a successive Interest at time $t = 30ms$, which will reach router r_4 at time $t = 60ms$.

Fig. 4.13 and 4.14 illustrate the mean AF and SD values of an approach with regard to the catalog size $|O|$ and the distance of a content source from the ISP/AS network in hop-counts hc , when caching is either disabled or enabled, and $|O| \in \{10^6, 10^8\}$ and $hc \in \{1, 3, 5\}$. The purpose of this evaluation is to highlight the impact of on-path caching on the performance of the mechanisms in question and a request aggregation mechanism. Referring to both figures, the following observations may be summarised:

- CS-ERA outperforms ORIGINAL by 11-15% and by 13-17% with regard to the AF metric, when caching is disabled and enabled, respectively, depending on the network topology and the parameters used.
- The SD values of the AF metric of both approaches are between 4-6%.

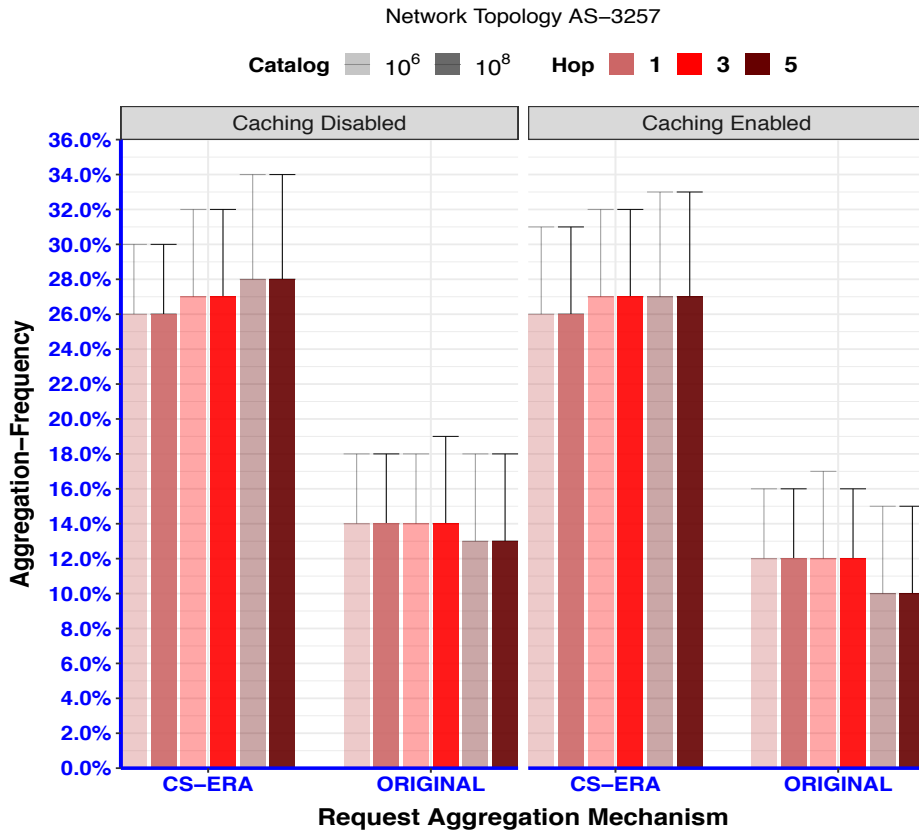


Fig. 4.13: Representation of the mean AF and SD values of an approach for the AS-3257 network topology, when caching is either disabled or enabled and $|O| = \{10^6, 10^8\}$ and $hc \in \{1, 3, 4\}$.

- On-path caching is more likely to affect the AF metric of ORIGINAL compared to CS-ERA, since the functionality of ORIGINAL is more vulnerable to the changes applied on the system model. More specifically, a reduction of 2-3% is concluded for ORIGINAL when caching is enabled, compared to the equivalent AF values when caching is disabled, while this difference is equal to 1% for CS-ERA at maximum.
- The reduction in the performance of the mechanisms in question and the request aggregation mechanism supported has been shown to be comparable to the benefit

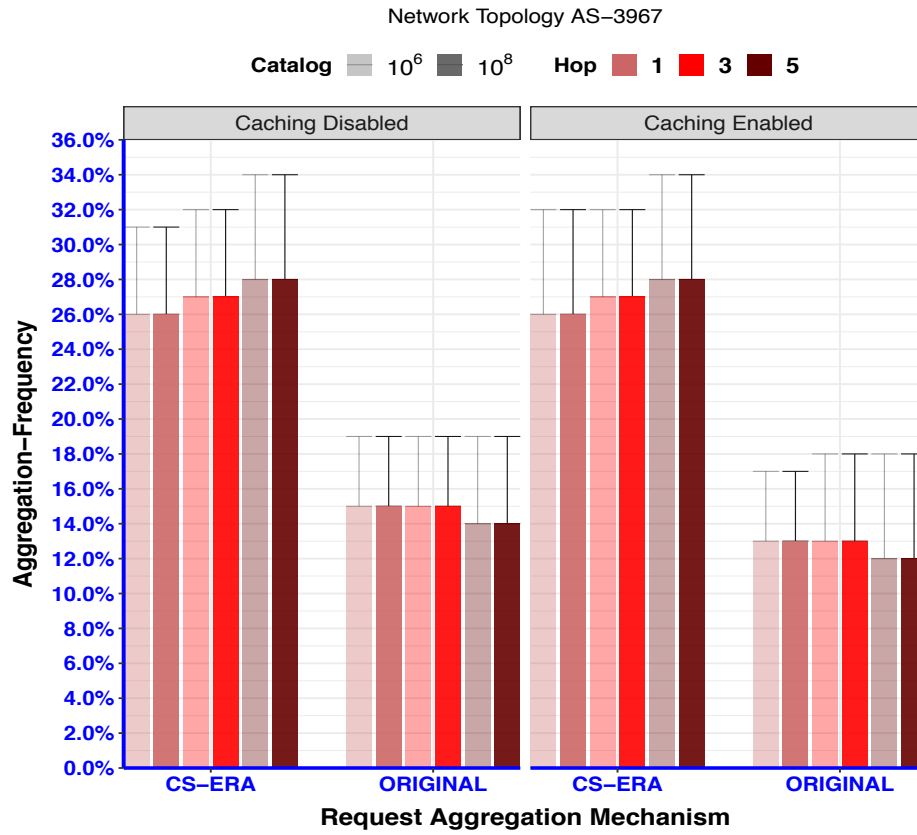


Fig. 4.14: Representation of the mean AF and SD values of an approach for the AS-3967 network topology, when caching is either disabled or enabled and $|O| = \{10^6, 10^8\}$ and $hc \in \{1, 3, 4\}$.

of LCE concluded in Section 2.3.7. The performance of LCE for the equivalent cache size, when single-Interest object resources are used, has been proven to be equal to 2% with regard to the cache-hits ratio (CHR) metric for either network topology.

Fig. 4.15 and 4.16 illustrate the mean AER and SD values of an approach with regard to the catalog size $|O|$ and the distance of a content source from the ISP/AS network in hop-counts hc , when caching is either disabled or enabled, and $|O| \in \{10^6, 10^8\}$ and

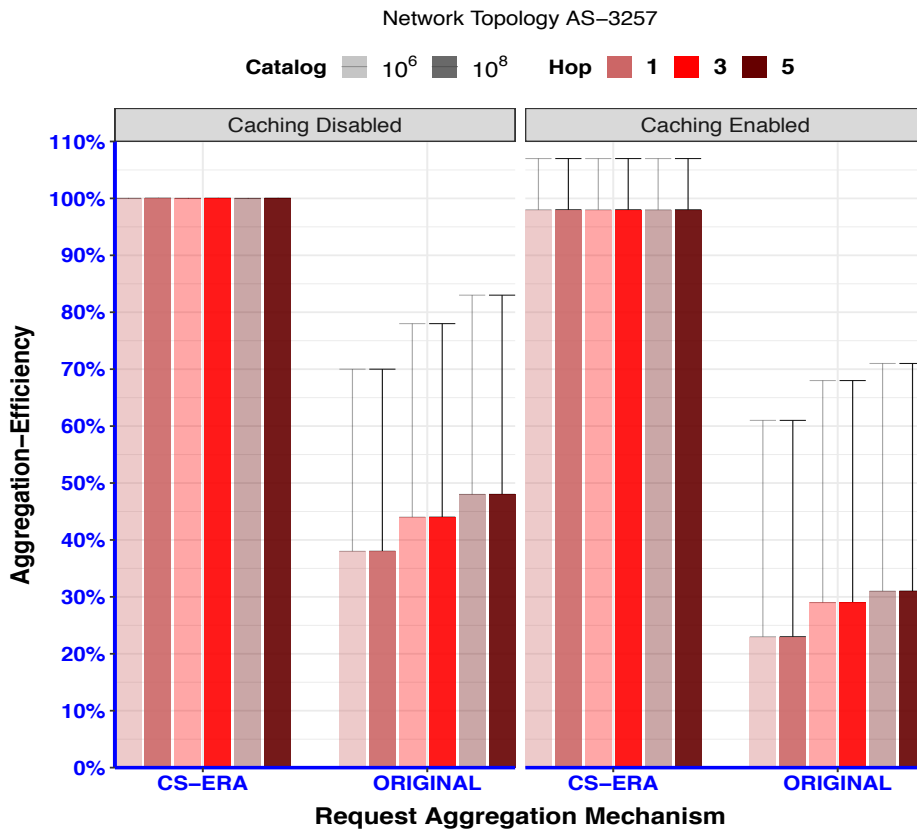


Fig. 4.15: Representation of the mean AER and SD values of an approach for the AS-3257 network topology, when caching is either disabled or enabled and $|O| = \{10^6, 10^8\}$ and $h \in \{1, 3, 4\}$.

$h \in \{1, 3, 5\}$. Referring to both figures, the following observations may be summarised:

- CS-ERA outperforms ORIGINAL by 52-68% and by 67-81% with regard to the AER metric, when caching is disabled and enabled, respectively, depending on the network topology and the parameters used.
- The SD values of the AER metric of both approaches have been shown to increase, since on-path caching affects the performance of request aggregation. The SD val-

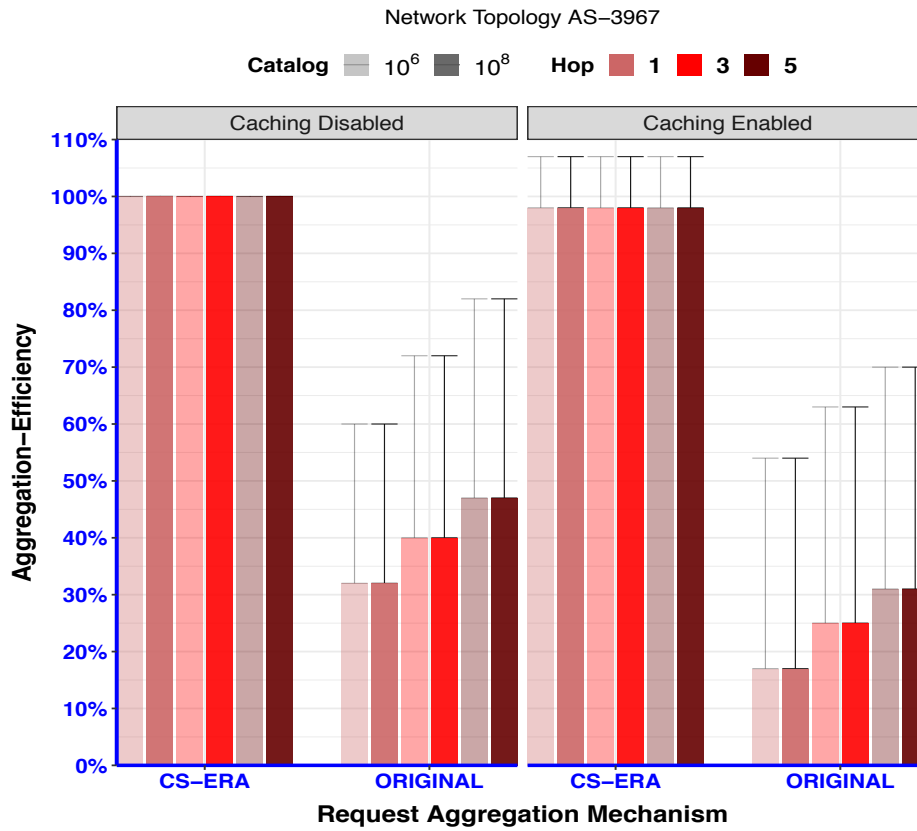


Fig. 4.16: Representation of the mean AER and SD values of an approach for the AS-3967 network topology, when caching is either disabled or enabled and $|O| = \{10^6, 10^8\}$ and $hce \in \{1, 3, 4\}$.

ues of ORIGINAL have increased by 4-9%, following the pattern of the equivalent mean AER values. The SD values of CS-ERA have increased by 9%, regardless of any parameters, indicating a higher variance of the metric.

- On-path caching is more likely to affect the AER metric of ORIGINAL compared to CS-ERA, since the functionality of ORIGINAL is more vulnerable to the changes applied on the system model. More specifically, a reduction between 15-17% is concluded for ORIGINAL when caching is enabled, compared to the equivalent

AER values when caching is disabled. This difference is equal to 2% for CS-ERA.

- Similar to the impact of the catalog size on the AF and AER metrics for either mechanism when caching is disabled, an equivalent impact is observed for the AF and AER metrics when caching is enabled.

4.5 Discussion

The purpose of this section is to provide a discussion on the effect of the design of CS-ERA on the performance of a request aggregation mechanism, while highlighting the limitations derived from its design, including the security implications and the deployment implications. Moreover, the discussion aims to identify the effect of the evaluation parameters used on the performance of the mechanisms in question and to conclude to the approach that yields to the highest performance gain.

4.5.1 Lessons Learned

In this section, a number of conclusions are summarised with regard to the performance of the mechanisms in question, listed as follows:

- CS-ERA focuses on content sharing between consumers that request to retrieve the same object resource. For this purpose, CS-ERA applies request aggregation on an object naming granularity rather than a chunk naming granularity. Due to these reasons, CS-ERA concludes to a higher performance benefit faster than ORIGINAL, while being more resilient with regard to the impact of on-path caching.
- An increase in the number of routers on the forwarding paths has been shown to benefit the performance of ORIGINAL by 1-2% for the AF metric, yet degrade its

performance by up to 6% for the AER metric. A higher number of routers on the forwarding paths increases the probability of request aggregation occurring within a network, yet it also increases the occurrence of single request aggregations. This term is used to refer to request aggregations of Interests that may occur at some point during the retrieval of an object resource, yet they will not occur again.

- An increase in the number of hops that a content source is placed away from the local ISP/AS network hc , is expected to benefit the performance of either mechanism for either evaluation metric, i.e. AF or AER.
- CS-ERA is more resilient with regard to the changes applied on either evaluation parameter, compared to ORIGINAL. This is due to the fact that CS-ERA focuses on aggregating a higher range of Interests for an object resource, while once request aggregation occurs CS-ERA bounds the successive Interests issued for the same object resource to be satisfied locally within boundaries of the ISP/AS network.
- Due to the functionality of CS-ERA, CS-ERA is able to satisfy almost all Interests issued for an object resource locally within the ISP/AS network. On the contrary, ORIGINAL is able to satisfy up to 48% of the same Interests locally.
- CS-ERA outperforms ORIGINAL by 11-15% and by 52-68% for the AF and AER metrics when caching is disabled. CS-ERA outperforms ORIGINAL by 13-17% and 67-81% with regard to the same evaluation metrics, when caching is enabled.
- Either mechanism has been shown to be affected by the feature of on-path caching and the performance of LCE, while the difference on the performance of CS-ERA against ORIGINAL is highlighted when caching is enabled.

The conclusions presented in this section highlight the superiority of CS-ERA and the benefits derived from its functionality and the design choices made.

4.5.2 Evaluation Limitations

Following the discussion on the evaluation limitations described in Sections 2.3.9 and 3.5.2, with regard to the system model adopted in the evaluation of a number of caching policies and PLbC, a discussion on the evaluation limitations of the original request aggregation mechanism and the mechanism of CS-ERA is presented in this section.

Similar to the evaluation of caching policies, the evaluation of both the original request aggregation mechanism and the mechanism of CS-ERA has been limited to the AS-3257 network topology and the AS-3967 network topology provided by the ndnSIM community for experimentation. These are relatively small ISP/AS network topologies compared to the Rocketfuel topologies [Spring] and the CAIDA topologies [Caida]. Consequently, even though the results concluded from such network topologies may provide us with an intuition about the performance of the mechanisms in question, they should not be generalised. To this end, a variety of network topologies, including large network topologies [Chai , Wang 14] and disconnected network topologies [Badov] should be considered. Section 4.5.3 provides additional information of the network topologies to be considered to further define the benefits of CS-ERA. Consulting Section 4.4.2, an increase in the number of routers on the forwarding and the delivery paths expected in larger network topologies is not expected to affect the performance of CS-ERA. An increase of the same parameter is expected to increase the occurrence of single request aggregations for the original request aggregation mechanism, degrading its performance. Moreover, highly connected network topologies, such as the AS-3257 network topology and the AS-3967 network topology will benefit the performance of either mechanism

since more forwarding paths will be merged within a network, increasing the probability of request aggregation occurring. In addition to this, and similar to the impact expected on the performance of caching policies from network topologies where the link capacity and the delay introduced is heterogenous, the same parameters are expected to affect the performance of the original request aggregation mechanism and the mechanism of CS-ERA. As an example, a lower benefit is expected to be concluded for CS-ERA in network topologies where the edge routers are connected to lower capacity links than the backbone routers, compared to network topologies where the link capacity is homogeneous.

Another evaluation parameter expected to affect the performance of the original request aggregation mechanism and the mechanism of CS-ERA is the content popularity distribution. A content popularity distribution is used to define the object resource to be requested by a consumer at a given time. Highly skewed distributions, such as the Weibull distribution adopted in the evaluation presented in Section 4.4 favour the occurrence of request aggregations and the performance of either mechanism. This claim may be verified based on the fact that a higher number of requests will refer to a particular object resource at a given time. To this end, an interesting question is the amount of benefit concluded for either mechanism using alternative popularity distributions, such as the Zipf distribution and the Zipf-Mandelbrot distribution [Pentikousis].

Besides the network topology and the popularity distribution, the catalog size adopted in the evaluation of the original request aggregation mechanism and the mechanism of CS-ERA is also expected to affect the results. Even though Section 4.4.2 concludes that the difference between the catalog sizes adopted has been negligible to affect the performance of the mechanisms, a higher catalog size is expected to lessen their benefits [Dehghan]. A higher catalog size is also expected to highlight the performance differences between the mechanisms and the ability of CS-ERA to aggregate more re-

quests for an object resource. This claim may be explained based on the fact that a lower number of requests is expected to be issued for an object resource at a given time.

4.5.3 Design Limitations

By extending the naming granularity to which request aggregation is applied from a chunk to an object resource and by bounding successive Interests to be satisfied locally within an ISP/AS network, CS-ERA has been shown to outperform the original request aggregation mechanism and to reduce the inter-network traffic. Besides the performance benefits obtained from its design, CS-ERA may result in a number of limitations. The functionality of CS-ERA has been based on the assumption that only a principal consumer(s) may retrieve an object resource from a content source(s) located beyond the boundaries of the local ISP/AS network at a given time. Subordinate consumers may retrieve an object resource advertised by a principal consumer locally within the /AS network. Due to this design convention, CS-ERA creates single points of failure that may impact the experience of subordinate consumers. Referring to Section 4.2.4, and the upstream process, a single Interest is propagated upstream to a content source for the maximum sequence number seq of the Interests aggregated by CS-ERA. Assuming that this Interest or the corresponding Data is lost, an additional delay is to be experienced by both the principal consumer and the subordinate consumers. Although, propagating less Interests beyond the boundaries of the local network is expected to reduce failures, such as failures occurred due to congestion, this is not guaranteed. NDN natively overcomes network failures by enabling the routers and the consumers within a network to re-issue Interests for the missing chunks of an object resource upon the expiration of a Round-Trip Timer (RTT) [Yi a]. This timer-based approach may be time intensive. To this end, an Interest NACK packet may be sent downstream to inform the routers and

the consumers about the failure occurred, so they can determine their future action.

More into the design of CS-ERA and its limitations, CS-ERA requires the advertisement of temporary replicas of an object resource by a principal consumer(s) within the local ISP/AS network using the NLSR protocol [Hoque]. This process is expected to introduce an additional overhead within the network that may delay the convergence of FIB tables and increase the content delivery times experienced by subordinate consumers. Although this overhead is expected to occur for the very first Interests requested locally within a network, this is not guaranteed. For this reason, the LFBR forwarding strategy has been designed to ensure that an Interest will reach either a temporary cache that possesses a replica of the content, or a content source. This in turn ensures that consumers are able to retrieve content and recover from cases where the FIB entries of NDN routers have not been populated yet, or they are invalid.

Besides LFBR, a number of modifications may be applied to the design of CS-ERA to ensure that such overheads will not affect the experience of subordinate consumers. An example of such modification is the propagation of Interest NACK packets to subordinate consumers to bound their successive Interests to be satisfied locally within the ISP/AS network. This design choice could be omitted since no guarantee exists that a valid FIB entry will be available by the time the subordinate consumers have issued their Interests. A CS-ERA border router could apply the functionality of a request aggregation mechanism on a chunk naming granularity, while maintaining the maximum number of chunks that a consumer has retrieved and whose Interests have been aggregated. This information will allow a CS-ERA border router to determine potential principal consumers. Once a consumer has been informed to act as a principal consumer and the entries for the given prefix have been established within the FIBs of the NDN routers, the LFBR forwarding strategy will be able to exploit the local replica.

4.5.4 Deployment Implications

Further to Section 4.5.3, and the design limitations derived from the design of CS-ERA, a number of practical limitations may be encountered with regard to the adaptation and the deployment of this mechanism within an ISP/AS network. More specifically, the adaptation of the NLSR protocol by principal consumers to advertise temporary replicas of an object resource within the local ISP/AS network may introduce a significant traffic overhead. The amount of this overhead depends on the number of request aggregations occurred at CS-ERA border routers and the number of consumers instructed to act as principal consumers. Both factors are dependent to a number of parameters including the network topology, the number of consumers and the content popularity distribution. A higher number of principal consumers is expected to increase the number of Prefix LSAs to be exchanged between the NDN routers within an ISP/AS network. Given the dynamic environment of ICN and the convergence challenges introduced by the NLSR protocol [Hoque], an alternative to NLSR may be preferable: A name resolution service may be used specifically for the purpose of advertising temporary replicas of object resources within an ISP/AS network [Ahlgren b, Xylomenos]. Name resolution services necessitate a centralised configuration module that maintains full knowledge of the network topology and the routers within a network. Nonetheless, centralised modules are expected to provide a faster and more accurate representation of the network and the content availability of principal consumers at a given time.

An additional practical limitation that may be encountered with regard to the adaptation and the deployment of CS-ERA within an ISP/AS network is the availability of consumers to act as principal consumers. For this purpose, an important prerequisite is the availability of the cache capacity of consumers to store temporary replicas, which

is expected to be limited. Although this practical limitation may not be applicable to private business ASes in which all consumers may be instructed to share a portion of their cache capacity within the local network, commercial ISP/AS networks may not be as simple to manage. To overcome this limitation, a software tool may be used, through which consumers may define their availability and the portion of their cache capacity to be shared within the network [Zhu].

Another practical implication that may be encountered is the management of the load introduced to principal consumers, generated by subordinate consumers through the exploitation of the LFBR forwarding strategy to retrieve an object resource within the local ISP/AS network, if a local replica exists. Currently, CS-ERA does not support any load management capabilities. Nonetheless, such information may be easily extracted from the CS-ERA border routers. For this purpose, CS-ERA border routers could be instructed to collect information with regard to the number of consumers whose Interests have been aggregated and which have been instructed to act either as subordinate or as principal consumers. By periodically exchanging this type of information, CS-ERA border routers may cooperate to make optimal decisions on whether a consumer should be instructed to act as a subordinate or a principal consumer. Moreover, the LFBR forwarding strategy could be enriched with additional attributes to introduce load balancing functionalities within an ISP/AS network and to prevent subordinate consumers from overloading a particular subset of principal consumers.

4.5.5 Security Implications

Similar to the security implications with regard to the design of the PLbC caching policy, a number of security implications may be derived from the design of CS-ERA, which refer to the integrity of the Data packet format and the raise of potential *FIB pollution*

attacks [Mannes]. More specifically, the `AdvertiseReplica` field of the Data packet format could be modified, either by an intermediate adversary or a malicious router, to enforce consumers to act as principal consumers and to advertise their replicas within the local ISP/AS network using the NLSR protocol. This is expected to increase both the network traffic and the content delivery times of consumers since a significant number of LSAs will flood the network. At this point it is worth noting, that the NLSR protocol has no prior knowledge on whether a consumer should act as a principal consumer.

Similar to the security measures described for PLbC, a lightweight hop-by-hop encryption algorithm, e.g. an SHA algorithm, could be used towards ensuring that the fields of a Data packet will remain intact. In contrast to PLbC, which requires the application of an encryption algorithm at all routers from a content source to consumers, CS-ERA requires the application of this technique at the routers from the CS-ERA border router to the local router of a potential principal consumer. Nonetheless, an SHA algorithm is not sufficient to ensure the integrity of a Data packet. To this end, an additional mechanism may be used, where local routers may refer to border routers about whether a prefix should be advertised within the local network. For this purpose, CS-ERA border routers are expected to keep track of the most recent prefixes for which a Data packet with an enabled `AdvertiseReplica` field has been sent downstream. If the reply is positive, the LSA will be propagated further. If otherwise, the LSA will be dropped. To further identify the origin of this attack, as well as malicious routers, a local router may distribute a push-back alert to the upstream routers that a potential attack may be in progress. This push-back technique may also be used to avoid malicious routers on delivery paths [Gasti, Yi a].

Besides the aforementioned security implications, an additional implication related to the design of CS-ERA is the identification of a consumer acting as a temporary content

source for a given prefix, which may be considered as a privacy concern. This type of information may be obtained from the name and AdvertiseReplica fields of a Data packet. Although observing these fields may expose some information with regard to the content to be advertised and the location of a principal consumer, the same information may be obtained by monitoring Interest packets in the current design of the NDN architecture.

4.5.6 Request Aggregation vs. On-path Caching

On-path caching has been proposed as a feature of the NDN architecture that allows the routers on delivery paths to cache chunks of an object resource for a short period of time. On-path caching allows successive Interests that refer to the same cached content to be satisfied locally within an ISP/AS network, thus reduce the traffic propagated within the local network and beyond the boundaries of this network to the Internet infrastructure. In addition to the reduced network traffic, on-path caching is expected to reduce the content delivery times of consumers by shortening the distance to be traversed between consumers and the potential content sources.

Request aggregation is a flow-control mechanism and a security mechanism of the NDN architecture, that merges successive Interests that refer to the same chunk of an object resource and allows only one of them to be propagated to a content source(s). Similar to on-path caching, request aggregation is expected to reduce the traffic propagated within the local network and beyond the boundaries of this network to the Internet infrastructure. Moreover, request aggregation has been shown to result in a reduction of the distance between consumers and the potential content sources, hence reduce the content delivery times of consumers [Dabirmoghaddam ,Dehghan].

Referring to Section 1.3, and the objectives of this thesis, both the on-path caching feature and the request aggregation mechanism may be exploited to satisfy the objec-

tives stated. Nonetheless, according to a number of studies focused on the analysis of the efficiency of a request aggregation mechanism and the impact of the feature of on-path caching on its performance [Dabirmoghaddam , Ghali], the efficiency of a request aggregation mechanism has been proven to be negatively affected by the performance of a caching policy. This outcome may be explained as follows: Since, on-path caching allows Interests to be satisfied locally at a router and prevents them from being propagated upstream to a content source(s), on-path caching cancels the benefit of a request aggregation mechanism for the content being cached. Hence, the efficiency of a request aggregation mechanism depends on the performance of a caching policy. This conclusion has been verified based on simulations and analytical models in the same studies, and based on simulations in the study presented in this chapter.

Yet, different studies have suggested that request aggregation may operate complementary to on-path caching, increasing the overall network performance [Shannigrahi]. This contradiction depends on the chunk size of the traffic model adopted, i.e. equal to 100MB, compared to the default chunk size defined, i.e. equal to 10KB. This increase on the chunk size suggests an increase on the lifetime of PIT entries by a magnitude of 10^4 , which in turn increases the probability of request aggregation occurring within an NDN network. The conclusions indicate that a certain type of use-case scenarios may exist, where an on-path caching policy and a request aggregation mechanism may cooperate to enhance the performance of a network and reduce the content delivery times of consumers. Moreover, the co-existence of a caching policy and a request aggregation mechanism at an NDN router is not necessary; either of these mechanisms may be disabled or not supported. To this end, a summary of potential use-case scenarios where the feature of on-path caching and the mechanism of request aggregation may either work complementary, or in isolation to one another, is given as follows:

- When the cost of maintaining in-network caches, able to reduce the intra and inter-network traffic of a local ISP/AS network is prohibitable. Examples of such use-case scenarios are small organisations that aim to maintain their own network infrastructure due to security reasons, such as a university campus. In these scenarios, the mechanism of request aggregation is expected to be beneficial.
- When the catalog size or the size of popular content requested within a local ISP/AS network is much higher to the size of content that caches may maintain to preserve acceptable CS lookup times (Section 1.3.1), or when a reduction in the cache size of caches is of economical interest. An example of such use-case scenario is social media content, whose catalog sizes rise to the size of petabytes [Borthakur]. In these scenarios, both the on-path caching feature and the mechanism of request aggregation are expected to be beneficial.
- When the chunk size of object resources is much higher to the default chunk size of an object resource, i.e. equal to 10KB. A higher chunk size is expected to degrade the performance of a cache since a higher cache capacity will be occupied by a smaller group of contents, preventing more content to be cached [Arlitt]. On the contrary, higher chunk sizes have been shown to increase the efficiency of a request aggregation mechanism, since entries tend to stay in a PIT for a longer period of time. An example of such use-case scenario is the one presented in [Shannigrahi], of a real-world application that distributes scientific data. In these scenarios, the mechanism of request aggregation is expected to be beneficial, while the feature of on-path caching may be used to cache smaller chunks of content.
- When the content distributed within a local ISP/AS network is either not cacheable or not-beneficial to be cached. Examples of such use-case scenarios are sensitive

content whose name may have been encrypted by its own producer to ensure the content is not-cacheable [DiBenedetto], live-streaming content or content with short lifetimes, such as dynamic content [Oran]. In these scenarios, the mechanism of request aggregation is expected to be beneficial.

4.6 Summary

In this chapter, a mechanism that focuses on content sharing between consumers that request to retrieve the same object resource, called Content Sharing-Extended Request Aggregation (CS-ERA) mechanism, has been proposed. For this purpose, CS-ERA exploits the mechanism of request aggregation to bound successive requests to be satisfied within the boundaries of the local ISP/AS network by consumers that possess a replica, if one exists. This is achieved by modifying the Interest and Data packets to include information that will allow the communication between consumers within the same ISP/AS network and the satisfaction of Interests. To propagate Interests to a replica within the boundaries of the local network, an extended Best Route (BR) forwarding strategy has been adopted, called Locality-First Best Route (LFBR). To increase the number of successive requests to be bounded, CS-ERA extends the naming granularity to which request aggregation is applied from a chunk to an object resource. By doing so, CS-ERA increases the probability of request aggregation occurring within a NDN network, since any Interest that refers to either chunk of an object resource may be aggregated instead of the Interests that refer to a single chunk of an object resource at a given time.

CS-ERA has been evaluated against the original request aggregation mechanism, using the evaluation metrics of AF and AER, and a number of evaluation parameters, when caching is either disabled or enabled. According to the results, CS-ERA has been

shown to outperform ORIGINAL by 11-15% with regard to the AF metric, and by 52-68% with regard to the AER metric, when caching is disabled. CS-ERA has been shown to outperform ORIGINAL by 13-17% and 67-81% with regard to the same evaluation metrics, when caching is enabled. Even though, both mechanisms have been shown to be affected by the feature of on-path caching, the performance of CS-ERA has been shown to be less affected compared to ORIGINAL. Besides the benefits of CS-ERA, a number of limitations worth further investigation have been identified, such as the overhead introduced by the NLSR protocol used by a principal consumer(s) to advertise its temporary replica of an object resource(s) within the local ISP/AS network.

Chapter 5

Conclusions and Future Work

In this chapter, a summary of the contributions of this thesis is given in relation to the research objectives stated. For ease of understanding, the objectives of this thesis are presented, followed by a description of the solutions proposed to meet the objectives: The Probability and Location-based Caching (PLbC) policy and the Content Sharing-Extended Request Aggregation (CS-ERA) mechanism. The conclusions drawn from the evaluation of PLbC and CS-ERA against a number of caching policies and the original request aggregation mechanism, respectively, are given. A number of suggestions for improvements and future work are proposed for both mechanisms.

5.1 Objectives and Summary

The problem domain of this thesis has been defined as the exploration of the on-path caching feature and the request aggregation mechanism of the NDN ICN architecture to enhance the network performance and the consumer experience within an ISP/AS network. For this purpose the following two objectives have been defined: The reduction

of the network traffic within an ISP/AS network and beyond the boundaries of this network to the Internet infrastructure, and the reduction of the content delivery times of consumers. By reducing the intra and inter-network traffic, the content delivery times experienced by consumers are also expected to be reduced since each Interest-Data pair will need to traverse a lower number of hops. Consequently, the reduction of the intra and inter-network traffic is considered to be the main objective of this thesis, while enhancing the consumer experience is considered to be complementary to the first objective.

In order to address the objectives stated, two orthogonal approaches have been investigated: A dynamic-probability caching policy, called *Popularity and Location-based Caching (PLbC)*, and a mechanism that focuses on content sharing between consumers within an ISP/AS network by exploiting the request aggregation mechanism of NDN, called *Content Sharing-Extended Request Aggregation (CS-ERA)*.

5.1.1 PLbC

PLbC is a lightweight caching algorithm that aims to cache popular content closer to consumers. By doing so, PLbC aims to reduce the traffic generated within a local ISP/AS network due to the propagation of both Interests and Data. Moreover, PLbC aims to reduce the content delivery times of consumers, since each hop traversed by the Interest and Data packets adds to the content delivery times experienced by consumers. For this purpose, PLbC utilises both the criterion of content popularity and the criterion of the location of routers on delivery paths by incorporating them into the caching decision process to construct a probability that aims to determine the benefit of caching content locally at an NDN router. To determine the popularity of content, PLbC exploits the information maintained in the Statistics Information Table (SIT) of an NDN router with regard to the object requests received. To determine the location of routers, PLbC

exploits the information maintained in the Interest and Data packet formats with regard to the number of hops traversed on the forwarding and the delivery path(s), respectively.

PLbC has been evaluated against the caching policy that has been proven to perform the best among a number of caching policies evaluated in this thesis, i.e. ProbCache+ (PC+). PLbC has been shown to yield a 3-6% higher probability of retrieving the content locally from the caches of routers compared to PC+, while caching 11-15% less content at the routers within an ISP/AS network. Yet, PLbC is unable to fully utilise the cache capacity of routers when their size approaches the catalog size. The reason for this complication is the low values concluded by the correlativity-based content popularity factor P_f . At this point it is worth noting, that such cache sizes, e.g. equal to 100GBs in real deployments, are expected to introduce a significant computational overhead at routers, increasing both the CS lookup time and the content delivery times of consumers. Hence, even though this conclusion is alarming for the performance of PLbC, its application is questionable given the computational overhead that such cache sizes are expected to introduce. The evaluation of PLbC has been based on a number of evaluation parameters such as network topologies, cache sizes and catalog sizes.

5.1.2 CS-ERA

CS-ERA is a mechanism that focuses on content sharing between consumers that request to retrieve the same object resource within an ISP/AS network. For this purpose, CS-ERA exploits the mechanism of request aggregation of the NDN architecture and the information in a PIT to bound successive Interests of consumers interested in the same object resource to be satisfied locally within the ISP/AS network, if a local replica exists. By doing so, CS-ERA aims to reduce the network traffic generated within the local ISP/AS network due to the propagation of both Interests and Data and the traffic

propagated beyond the boundaries of this network to the Internet infrastructure. By reducing the intra and inter-network traffic, CS-ERA is also expected to reduce the content delivery times of consumers, assuming that each hop traversed by the Interest and Data packets beyond the local ISP/AS network adds a notable overhead to the content delivery times. To increase the number of Interests to be bounded, CS-ERA extends the naming granularity to which request aggregation is applied from a chunk to an object resource, allowing any Interest that refers to either chunk of an object resource to be aggregated at a given time. To enable the communication between consumers within the same ISP/AS network and the satisfaction of successive Interests, CS-ERA exploits the information maintained in the Interest and Data packets. To propagate Interests to a replica within the local ISP/AS network, an extended Best Route (BR) forwarding strategy has been adopted, called *Locality-First Best-Route (LFBR)*.

CS-ERA has been evaluated against the original request aggregation mechanism. CS-ERA has been shown to outperform the original request aggregation mechanism by aggregating 11-15% more object requests within an ISP/AS network, while satisfying 52-68% more chunks of an object resource, when caching is disabled. The same values when caching is enabled are equal to 13-17% and 67-81%, respectively. The evaluation of CS-ERA has been based on a number of evaluation parameters such as network topologies, catalog sizes and the distance of a content source from the local ISP/AS network.

Given the research objectives stated in this thesis and according to the evaluation results described for both PLbC and CS-ERA, the following conclusion is made: *Both PLbC and CS-ERA have met the research objectives stated in this thesis.*

5.2 Contributions

In this thesis, both the on-path caching feature and the mechanism of request aggregation have been explored to satisfy the objectives stated above. Towards satisfying these objectives, a number of contributions have been made, summarised as follows:

- A categorisation of the existing caching policies in the context of on-path caching and a taxonomy that highlights the evolution of the caching policies with regard to the caching criteria used to perform a caching decision.
- An evaluation of a number of dynamic-probability caching policies and a number of benchmark caching policies based on simulations, that provides an intuition on the effect of the caching criteria used by a caching policy to perform a caching decision on the performance of a network and the experience of consumers. The conclusions drawn from this evaluation have been shown to verify the observations drawn from the taxonomy of caching policies. The evaluation also provides an intuition with regard to the effect of the evaluation parameters of network topology, cache size and catalog size on the performance of caching policies.
- A dynamic-probability caching policy, i.e. PLbC, constructed based on the information collected and the conclusions drawn from the taxonomy of the caching policies reviewed in this thesis and the evaluation of a number of dynamic-probability and benchmark caching policies. PLbC utilises both the criterion of content popularity and the criterion of the location of routers on delivery paths by incorporating them into the caching decision process to construct a probability. The evaluation of PLbC against a number of evaluation parameters has been shown to verify the intuitions for these caching criteria by reducing the intra-network traffic.

- A content sharing mechanism, i.e. CS-ERA, constructed to share content between consumers within the local ISP/AS network interested in the same object resource, through exploiting the request aggregation mechanism. CS-ERA extends the naming granularity to which request aggregation is applied from a chunk to an object, while bounding successive Interests to be satisfied locally within a network. The evaluation of CS-ERA against a number of evaluation parameters has shown that the probability of request aggregation occurring within an NDN network may indeed be increased. Moreover, CS-ERA has been shown to benefit the performance a network and the experience of consumers by reducing the inter-network traffic.
- A summary of the limitations, security implications and lessons learned from the design and the evaluation of PLbC and CS-ERA, that are expected to be beneficial to future researchers in the area of NDN and the wider ICN research community.
- A quantitative proof that the performance of a request aggregation mechanism is negatively affected by the performance of an on-path caching policy and the suggestion of use-case scenarios where the co-existence of these mechanisms may be beneficial, or where one may be preferable than the other due to certain limitations.
- A quantitative proof that either of the mechanisms proposed may be used to enhance the performance of a network and the experience of consumers, the objectives of this thesis.

5.3 Future Work

In this section, a number of suggestions and directions for future work based on the design, the evaluation and the performance of the mechanisms proposed is given.

5.3.1 PLbC

Referring to Section 3.5.1, and the conclusions drawn from the performance of PLbC against the evaluation parameters used, PLbC has been shown to be unable to fully utilise the cache capacity of routers when their size increases to 100GBs, due to the low values concluded by the correlativity-based content popularity factor P_f . Although such cache sizes are expected to increase both the CS lookup times and the content delivery times of consumers, a method where the values of the P_f factor do not result in an underutilisation of the cache capacity of routers needs to be defined. Consulting the performance of PC+ as the cache capacity of routers increases, one may conclude that the criterion of the cache capacity of routers should not be omitted from the design of an on-path caching policy. To this end, PLbC could be modified to allow not-so-popular content to be cached when unutilised cache capacity exists. This may be achieved by consulting the cache performance rates of a router in consecutive time intervals. More specifically, if unutilised cache capacity exists, a router may decide to lessen its rule of caching popular content by comparing the popularity of a content against the most popular content to the second most popular content. A router may then observe its performance rates to decide whether caching not-so-popular content has benefit these rates. Based on this comparison, a router may adjust its behaviour accordingly and decide whether to fall-back to its former caching rule or to lessen the comparison further.

More into the suggestions and directions for future work, an evaluation of the performance of PLbC that takes into account real-time parameters is suggested [Project c], to evaluate the effect of the size of an SIT table and the overhead introduced by it at an NDN router. An SIT table is used to maintain the popularity counters ctr_o of object resources used to decided the popularity of content. The size of this table is

expected to grow linearly to the number of individual objects requested within an ISP/AS network. To avoid the construction of large SIT tables, a lifetime is bound to an SIT entry to enforce the collection of statistical information about the object resources for which the requesting activity remains active. A lifetime is initialised and updated upon the arrival of an object request. Hence, a lifetime is related to the network traffic patterns of the local ISP/AS network and should be defined a priori by an ISP. Nonetheless, and according to former studies that aim to predict the popularity of UGC content [Ahmed , Cha , Figueiredo], a lifetime of a few hours, e.g. between half an hour and 2 hours, may be suggested to capture the popularity of contents. At this point it is worth noting, that translating the aforementioned lifetimes into simulation time may result into invalid conclusions since no scientific method exists to support this conversion. As an example, Psaras et al. in [Psaras a] used a number of experimental time intervals to conclude that the performance of their algorithm works better for simulation times different to the real-times suggested consulting the existing technologies.

5.3.2 CS-ERA

Referring to Section 4.5, and the limitations of CS-ERA, CS-ERA is expected to create single points of failure that may impact the experience of consumers when frequent interruptions occur at the backbone network. Although propagating less Interests beyond the boundaries of the local ISP/AS network is expected to prevent the occurrence of failures, such as failures occurred due to congestion, this is not guaranteed. NDN natively overcomes network failures by enabling the routers and the consumers within a network to re-issue Interests for the missing chunks of an object resource upon the expiration of a RTT timer. Moreover, routers in NDN may propagate Interest NACK packets downstream to inform both routers and consumers about the failure, so they may

adjust their behaviour accordingly [Yi a]. An evaluation of the performance of CS-ERA over network environments where frequent interruptions occur is suggested, using either a timer-based approach and/or an Interest NACK approach to overcome the network failures experienced. This evaluation will allow the researcher to identify the resilience of this mechanism in such environments.

More into the suggestions and directions for future work, an evaluation of the overhead introduced by the NLSR protocol [Hoque] to advertise temporary replicas of an object resource within the local ISP/AS network, is suggested. Although, this overhead is expected to occur for the very first Interests requested locally within a network, this is not guaranteed. For this purpose, the design of CS-ERA may be modified to prevent the occurrence of such overheads affecting the experience of consumers. As an example CS-ERA border routers may omit bounding the Interests of consumers within the local ISP/AS network since no guarantee exists that a consumer has indeed accepted the request of a CS-ERA border router to act as a principal consumer. In addition to this, no guarantee exists that the corresponding FIB entries have already converged. As an alternative, the LFBR forwarding strategy may be used to exploit local replicas, since LFBR has been designed to propagate Interests through the local faces of a FIB entry instead of the global faces of the same entry, if local faces that lead to a local replica of the content exist. To further avoid any unnecessary overheads, the NLSR protocol may be modified to accept advertisements of local replicas from consumers that intend to serve the content for a long period of time. Besides the evaluation of NLSR, a number of alternative approaches may be considered to advertise temporary replicas within the local ISP/AS network. As an example, a name resolution service could be used [Ahlgren b, Xylomenos]. Name resolution services necessitate a centralised configuration module that maintains full knowledge of the network topology and the routers

within a network. Nonetheless, centralised modules are expected to provide a faster and more accurate representation of the network and the content availability of principal consumers at a given time.

Another direction for future work is the evaluation of the load introduced to principal consumers, generated by subordinate consumers through the exploitation of the LFBR forwarding strategy to retrieve an object resource locally within an ISP/AS network. This is achieved by instructing a number of consumers within an ISP/AS network to act as principal consumers and to advertise the availability of their temporary replicas. A number of consumers are also instructed to act as subordinate consumers and to bound their successive Interests to be satisfied within the local network by exploiting the temporary replicas of principal consumers. According to Section 4.5.4, CS-ERA does not currently support any load management capabilities. Consequently, this problem is open to further research and experimentation. As an example, CS-ERA border routers could be exploited to collect information about the number of consumers whose Interests have been aggregated and which have been instructed to act either as subordinate or as principal consumers. Based on this information, the network traffic and the load state of principal consumers, a decision on whether a consumer should be instructed to act as a subordinate or a principal consumer may be made. To further optimise these decisions, CS-ERA border routers may periodically exchange information to enforce cooperation within the boundaries of an ISP/AS network.

The use-case scenarios suggested in Section 4.5.6, where the co-existence of the feature of on-path caching and the mechanism of request aggregation may be beneficial to the performance of a network and the experience of consumers, or where one may be preferable than the other, are also potential directions for future work.

Bibliography

- [Abu] A.J. Abu, B. Bensaou & J.M. Wang. *Interest Packets Re-transmission in Lossy CCN Networks and its Impact on Network Performance*. In Proceedings of the ACM Conference on Information-Centric Networking (ICN'14), Paris, France, September 2014, pages 167–176.
- [Adhatarao] S.S. Adhatarao, J. Chen, M. Arumaithurai, X. Fu & K.K. Ramakrishnan. *Comparison of Naming Schema in ICN*. In Proceedings of the IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN'16), Rome, Italy, June 2016, pages 1–6.
- [Afanasyev a] A. Afanasyev. *How to Get the Transmission Delay, NDN Mailing List*. (May 2013), Retrieved from <https://www.lists.cs.ucla.edu/pipermail/ndnsim/2013-May/000542.html>.
- [Afanasyev b] A. Afanasyev. *Importing Rocketfuel Topologies and Script to Convert them to AnnotatedTopologyFormat*,

cawka/ndnSIM-sample-topologies. Retrieved May 2020 from <https://github.com/cawka/ndnSIM-sample-topologies>.

- [Afanasyev c] A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun & L. Zhang. *Interest Flooding Attack and Countermeasures in Named Data Networking*. In Proceedings of the International Federation for Information Processing Networking Conference (IFIP'13), Brooklyn, New York, USA, May 2013, pages 1–9.
- [Afanasyev d] A. Afanasyev, S. Mastorakis, I. Moiseenko & L. Zhang. *Examples, NS-3 Named Data Networking (NDN) simulator*. Retrieved May 2020 from <https://ndnsim.net/2.5/examples.html>.
- [Afanasyev e] A. Afanasyev, I. Moiseenko & L. Zhang. *ndnSIM: NDN Simulator for NS-3*. Rapport technique, NDN Technical Report, NDN-0005, Revision 2, October 2012.
- [Ahlgren a] B. Ahlgren, M. D'Ambrosio, M. Marchisio, I. Marsh, C. Dannewitz, B. Ohlman, K. Pentikousis, O. Strandberg, R. Rembarz & V. Vercellone. *Design Considerations for a Network of Information*. In Proceedings of the International Conference on Emerging Networking Experiments and Technologies (CoNEXT'08), Madrid, Spain, December 2008, pages 1–6.
- [Ahlgren b] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher & B. Ohlman. *A Survey of Information-Centric Networking*.

IEEE Communications Magazine, July 2012, vol. 50, no. 7, pages 26–36.

- [Ahmed] M. Ahmed, S. Spagna, F. Huici & S. Niccolini. *A Peek into the Future: Predicting the Evolution of Popularity in User Generated Content*. In Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM'13), Rome, Italy, February 2013, pages 1–10.
- [Anand] A. Anand, V. Sekar & A. Akella. *SmartRE: An Architecture for Coordinated Network-wide Redundancy Elimination*. In Proceedings of the ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM'09), Barcelona, Spain, August 2009, pages 87–98.
- [Androutsellis-Theotokis] S. Androutsellis-Theotokis & D. Spinellis. *A Survey of Peer-to-peer Content Distribution Technologies*. ACM Computing Surveys, December 2004, vol. 36, no. 4, pages 335–371.
- [Arianfar] S. Arianfar, P. Nikander & J. Ott. *On Content-Centric Router Design and Implications*. In Proceedings of the CoNEXT Workshop on Re-Architecting the Internet (ReARCH'10), Philadelphia, Pennsylvania, USA, November 2010, pages 1–6.
- [Arlitt] M. Arlitt, R. Friedrich & T. Jin. *Performance Evaluation of Web Proxy Cache Replacement Policies*. Elsevier Journal on

Performance Evaluation, February 2000, vol. 39, no. 1, pages 149–164.

[Badov] M. Badov, A. Seetharam, J. Kurose, V. Firoiu & S. Nanda. *Congestion-aware Caching and Search in Information-Centric Networks*. In Proceedings of the ACM Conference on Information-Centric Networking (ICN'14), Paris, France, September 2014, pages 37–46.

[Balakrishnan] H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S. Shenker, I. Stoica & M. Walfish. *A Layered Naming Architecture for the Internet*. In Proceedings of the ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM'04), Portland, Oregon, USA, August 2004, pages 343–352.

[Barabasi] A.L. Barabasi & R. Albert. *Emergence of Scaling in Random Networks*. Science Magazine, October 1999, vol. 286, no. 5439, pages 509–512.

[Bari] M.F. Bari, S.R. Chowdhury, R. Ahmed, R. Boutaba & B. Mathieu. *A Survey of Naming and Routing in Information-Centric Networks*. IEEE Communications Magazine, December 2012, vol. 50, no. 12, pages 44–53.

[Begtasevic] F. Begtasevic & P. Van Meighem. *Measurements of the Hopcount in Internet*. In Proceedings of the International Work-

shop on Passive and Active Measurement (PAM'01), Amsterdam, Netherlands, April 2001, pages 1–12.

- [Bernardini] C. Bernardini, T. Silverston & O. Festor. *MPC: Popularity-based Caching Strategy for Content Centric Networks*. In Proceedings of the IEEE International Conference on Communications (ICC'13), Budapest, Hungary, June 2013, pages 3619–3623.
- [Bollobas] B. Bollobas, O. Riordan, J. Spencer, G. Tusnady *et al.* *The Degree Sequence of a Scale-free Random Graph Process*. Wiley Journal on Random Structures and Algorithms, May 2001, vol. 18, no. 3, pages 279–290.
- [Borthakur] D. Borthakur. *Petabyte Scale Databases and Storage Systems at Facebook*. In Proceedings of the International Conference on Management of Data (SIGMOD'13), New York, New York, USA, June 2013, pages 1267–1268.
- [Buyya] R. Buyya, M. Pathan & A. Vakali. Content delivery networks. Springer Publishing Company Inc., 1st edition.
- [Caida] Caida. *Topology Research, Center for Applied Data Analysis*. (March 2019), Retrieved from <http://www.caida.org/research/topology/>.
- [Carofiglio a] G. Carofiglio, M. Gallo, L. Muscariello & D. Perino. *Modeling Data Transfer in Content-Centric Networking*. In Proceed-

ings of the International Teletraffic Congress (ITC'11), San Francisco, California, USA, September 2011, pages 111–118.

[Carofiglio b]

G. Carofiglio, L. Muscariello, M. Papalini, N. Rozhnova & X. Zeng. *Leveraging ICN In-network Control for Loss Detection and Recovery in Wireless Mobile Networks*. In Proceedings of the ACM Conference on Information-Centric Networking (ICN'16), Kyoto, Japan, September 2016, pages 50–59.

[Cha]

M. Cha, H. Kwak, P. Rodriguez, Y.Y. Ahn & S. Moon. *I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video system*. In Proceedings of the Internet Measurement Conference (IMC'07), San Diego, California, USA, October 2007, pages 1–14.

[Chai]

W. Chai, D. He, I. Psaras & G. Pavlou. *Cache "Less for More" in Information-Centric Networks*. In Proceedings of the International Federation for Information Processing Networking Conference (IFIP'12), Brooklyn, New York, USA, May 2012, pages 27–40.

[Chang]

H. Chang, S. Jamin & W. Willinger. *To Peer or not to Peer: Modeling the Evolution of the Internet's AS-level Topology*. In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'06), Barcelona, Spain, April 2006, pages 1–12.

- [Che] H. Che, Y. Tung & Z. Wang. *Hierarchical Web Caching Systems: Modeling, Design and Experimental Results*. IEEE Journal on Selected Areas in Communications, November 2002, vol. 20, no. 7, pages 1305–1314.
- [Cheng] X. Cheng, C. Dale & J. Liu. *Statistics and Social Network of YouYube Videos*. In Proceedings of the IEEE International Workshop on Quality of Service (IWQoS'08), Enschede, Netherlands, June 2008, pages 229–238.
- [Cheriton] D.R. Cheriton & M. Gritter. *TRIAD: A New Next-Generation Internet Architecture*. Stanford University, California, USA, January 2000.
- [Chiocchetti] R. Chiocchetti, D. Rossi & G. Rossini. *CcnSim: A Highly Scalable CCN Simulator*. In Proceedings of the IEEE International Conference on Communications (ICC'13), Budapest, Hungary, June 2013, pages 1–6.
- [Cho] K. Cho, M. Lee, K. Park, Kwon. T.T., Y. Choi & S. Pack. *WAVE: Popularity-based and Collaborative In-network Caching for Content-Oriented Networks*. In Proceedings of the IEEE INFOCOM Workshop on Emerging Design Choices in Named-Oriented Networking (NOMEN'12), Orlando, Florida, USA, March 2012, pages 1–6.
- [Choi] J. Choi, J. Han, E. Cho, T. Kwon & Y. Choi. *A Survey on Content-Oriented Networking for Efficient Content Delivery*.

IEEE Communications Magazine, March 2011, vol. 49, no. 3, pages 121–127.

- [Cisco] Cisco. *2016 Annual Report*. Rapport technique, Cisco Systems Inc., October 2016.
- [Compagno] A. Compagno, M. Conti, P. Gasti & G. Tsudik. *Poseidon: Mitigating Interest Flooding DDoS Attacks in Named Data Networking*. In Proceedings of the IEEE Conference on Local Computer Networks (LCN), Sydney, Australia, October 2013, pages 630–638.
- [Conti] M. Conti, P. Gasti & M. Teoli. *A Lightweight Mechanism for Detection of Cache Pollution Attacks in Named Data Networking*. Elsevier Journal on Computer Networks, September 2013, vol. 57, no. 16, pages 3178–3191.
- [Cooper] J.C.B. Cooper. *The Poisson and Exponential Distributions*. Mathematical Spectrum Magazine, 2005, vol. 37, no. 3, pages 123–125.
- [Dabirmoghaddam] A. Dabirmoghaddam, M. Dehghan & J.J. Garcia-Luna-Aceves. *Characterizing Interest Aggregation in Content-Centric Networks*. In Proceedings of the International Federation for Information Processing Networking Conference (IFIP’16), Vienna, Austria, May 2016, pages 449–457.
- [Dai] H. Dai, Y. Wang, J. Fan & B. Liu. *Mitigate DDoS Attacks in NDN by Interest Traceback*. In Proceedings of the IEEE IN-

FOCOM Workshop on Emerging Design Choices in Named-Oriented Networking (NOMEN'13), Turin, Italy, April 2013, pages 1–6.

[D'Ambrosio] M. D'Ambrosio, C. Dannewitz, H. Karl & V. Vercellone. *MDHT: A Hierarchical Name Resolution Service for Information-Centric Networks*. In Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking (ICN'11), Toronto, Ontario, Canada, August 2011, pages 7–12.

[Dannewitz] C. Dannewitz. *Network of Information (NetInf): An Information-Centric Networking Architecture*. Elsevier Journal on Computer Communications, April 2013, vol. 36, no. 7, pages 721–735.

[Dehghan] M. Dehghan, B. Jiang, A. Dabirmoghaddam & D. Towsley. *On the Analysis of Caches with Pending Interest Tables*. In Proceedings of the ACM Conference on Information-Centric Networking (ICN'15), San Francisco, California, USA, September 2015, pages 1–10.

[Detti a] A. Detti, N. Blefari Melazzi, S. Salsano & M. Pomposini. *CONET: A Content Centric Inter-Networking Architecture*. In Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking (ICN'11), August 2011, Toronto, Ontario, Canada, August 2011, pages 50–55.

- [Detti b] A. Detti, A. Caponi & N. Blefari-Melazzi. *Exploitation of Information Centric Networking Principles in Satellite Networks*. In Proceedings of the IEEE/AESS European Conference on Satellite Telecommunications (ESTEL'12), Rome, Italy, October 2012, pages 1–6.
- [Dhamdhere a] A. Dhamdhere & C. Dovrolis. *Can ISPs be Profitable without Violating "Network Neutrality"?* In Proceedings of the ACM SIGCOMM Workshop on Economics of Networked Systems (NetEcon'08), Seattle, Washington, USA, August 2008, pages 13–18.
- [Dhamdhere b] A. Dhamdhere & C. Dovrolis. *The Internet is Flat: Modeling the Transition from a Transit Hierarchy to a Peering Mesh*. In Proceedings of the International Conference on Emerging Networking Experiments and Technologies (CoNEXT'10), Philadelphia, Pennsylvania, USA, November 2010, pages 1–12.
- [DiBenedetto] S. DiBenedetto, P. Gasti, G. Tsudik & E. Uzun. *ANDANA: Anonymous Named Data Networking Application*. In Proceedings of the Annual Network and Distributed System Security Symposium (NDSS'12), San Diego, California, USA, February 2012, pages 1–18.
- [Domingues] G.M.B. Domingues, R.M.M. Leao & D.S. Mensche. *Enabling Information Centric Networks Through Opportunistic*

Search, Routing and Caching. Elsevier Journal on Computer Networks, June 2017, vol. 119, no. 3, pages 17–34.

[Fayazbakhsh] S.K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K.C. Ng, V. Sekar & S. Shenker. *Less Pain, Most of the Gain: Incrementally Deployable ICN*. In Proceedings of the ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM '13), Hong Kong, China, August 2013, pages 147–158.

[Fei] A. Fei, G. Pei, R. Liu & L. Zhang. *Measurements on Delay and Hop-Count of the Internet*. In Proceedings of the IEEE Global Communications Conference (GLOBECOM'98), Sydney, Australia, November 1998, pages 1–8.

[Feldmann] A. Feldmann. *Internet Clean-slate Design: What and Why?* ACM SIGCOMM Computer Communication Review, July 2007, vol. 37, no. 3, pages 59–64.

[Figueiredo] F. Figueiredo, F. Benevenuto & J.M. Almeida. *The Tube over Time: Characterizing Popularity Growth of YouTube Videos*. In Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM'11), Hong Kong, China, February 2011, pages 745–754.

[Fotiou] N. Fotiou, D. Trossen & G.C. Polyzos. *Illustrating a Publish-Subscribe Internet Architecture*. Springer Journal on

Telecommunication Systems, December 2012, vol. 51, no. 4, pages 233–245.

- [Gantz] J.F. Gantz. *The Expanding Digital Universe: A Forecast of Worldwide Information Growth Through 2010*. Rapport technique, White paper, IDC, March 2007.
- [Garcia] G. Garcia, A. Beben, F.J. Ramon, A. Maeso, I. Psaras, G. Pavlou, N. Wang, J. Sliwinski, S. Spirou & S. Soursos. *COMET: Content Mediator Architecture for Content-Aware Networks*. In Proceedings of the Future Network and Mobile Summit (FutureNetw'11), Warsaw, Poland, June 2011, pages 1–8.
- [Gasti] P. Gasti, G. Tsudik, E. Uzun & L. Zhang. *DoS and DDoS in Named Data Networking*. In Proceedings of the International Conference on Computer Communication and Networks (ICCCN'13), Nassau, Bahamas, July 2013, pages 1–7.
- [Ghali] C. Ghali, G. Tsudik, E. Uzun & C.A. Wood. *Living in a Pit-less World: A Case Against Stateful Forwarding in Content-Centric Networking*. Rapport technique, arxiv.org, Cornell University Library, New York, New York, USA, December 2015.
- [Ghodsi a] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti & S. Shenker. *Naming in Content-Oriented Architectures*. In Proceedings of the ACM SIGCOMM Workshop on

Information-Centric Networking (ICN'11), Toronto, Ontario, Canada, August 2011, pages 1–6.

[Ghodsii b] A. Ghodsii, S. Shenker, T. Koponen, A. Singla, B. Raghavan & J. Wilcox. *Information-Centric Networking: Seeing the Forest for the Trees*. In Proceedings of the ACM Workshop on Hot Topics in Networks (HotNet-X'11), Cambridge, Massachusetts, USA, November 2011, pages 1–6.

[Gill] P. Gill, M. Arlitt, Z. Li & A. Mahanti. *Youtube Traffic Characterization: A View from the Edge*. In Proceedings of the Internet Measurement Conference (IMC'07), San Diego, California, USA, October 2007, pages 15–28.

[Hassani] H. Hassani, M. Ghodsii & G. Howell. *A Note on Standard Deviation and Standard Error*. Oxford University Press, Journal on Teaching Mathematics and Its Applications, March 2010, no. 29, pages 108–112.

[Hoque] A.K.M.M. Hoque, S.O. Amin, A. Alyyan, B. Zhang, L. Zhang & L. Wang. *NLSR: Named-data Link State Routing Protocol*. In Proceedings of the ACM Conference on Information Centric-Networking (ICN'13), Hong Kong, China, August 2013, pages 15–20.

[Ioannou] A. Ioannou & S. Weber. *A Survey of Caching Policies and Forwarding Mechanisms in Information-Centric Networking*.

IEEE Communications Surveys and Tutorials, May 2016, vol. 18, no. 4, pages 2847 – 2886.

- [Jacobson] V. Jacobson, D.K. Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs & R.L. Braynard. *Networking Named Content*. In Proceedings of the International Conference on Emerging Networking Experiments and Technologies (CoNEXT'09), Rome, Italy, December 2009, pages 1–12.
- [Janaszka] T. Janaszka, D. Bursztynowski & M. Dzida. *On Popularity-based Load Balancing in Content Networks*. In Proceedings of the International Teletraffic Congress (ITC'12), Krakow, Poland, September 2012, pages 1–8.
- [Kamiyama] N. Kamiyama. *Analyzing Impact of Introducing CCN on Profit of ISPs*. IEEE Transactions on Network and Service Management, May 2015, vol. 12, no. 2, pages 176–187.
- [Kangasharju] J. Kangasharju, J. Roberts & K.W. Ross. *Object Replication Strategies in Content Distribution Networks*. Elsevier Journal on Computer Communications, March 2002, vol. 25, no. 4, pages 376–383.
- [Karami] A. Karami & M. Guerrero-Zapata. *An ANFIS-based Cache Replacement Method for Mitigating Cache Pollution Attacks in Named Data Networking*. Elsevier Journal on Computer Networks, April 2015, vol. 80, pages 51–65.

- [Kasiviswanathan] S. Kasiviswanathan, S. Eidenbenz & G. Yan. *Geography-based Structural Analysis of the Internet*. In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'11), Shanghai, China, April 2011, pages 1–9.
- [Katsaros] K.V. Katsaros, G. Xylomenos & G.C. Polyzos. *GlobeTraff: A Traffic Workload Generator for the Performance Evaluation of Future Internet Architectures*. In Proceedings of the IFIP International Conference on New Technologies, Mobility and Security (NTMS'12), Istanbul, Turkey, May 2012, pages 1–5.
- [Koponen] T. Koponen, M. Chawla, B.G. Chun, A. Ermolinskiy, K.H. Kim, S. Shenker & I. Stoica. *A Data-Oriented (and beyond) Network Architecture*. In Proceedings of the ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM'07), Kyoto, Japan, August 2007, pages 181–192.
- [Laoutaris] N. Laoutaris, H. Che & I. Stavrakakis. *The LCD Interconnection of LRU Caches and its Analysis*. Elsevier Journal on Performance Evaluation, July 2006, vol. 63, no. 7, pages 609–634.
- [Lee] U. Lee, I. Rimać & V. Hilt. *Greening the Internet with Content-Centric Networking*. In Proceedings of the International Conference on Energy-Efficient Computing and Net-

working (e-Energy'10), Passau, Germany, April 2010, pages 179–182.

- [Li a] B. Li, M.J. Golin, G.F. Italiano, X. Deng & K. Sohraby. *On the Optimal Placement of Web Proxies in the Internet*. In Proceedings of the IEEE International Conference on Communications (INFOCOM'19), New York, New York, USA, March 1999, pages 1282–1290.
- [Li b] J. Li, H. Wu, B. Liu, J. Lu, Y. Wang, X. Wang, Y. Zhang & L. Dong. *Popularity-driven Coordinated Caching in Named Data Networking*. In Proceedings of the ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS'12), Austin, Texas, USA, October 2012, pages 15–26.
- [Li c] W. S. Li, W. P. Hsiung, D. V. Kalashnikov, R. Sion, O. Po, D. Agrawal & K. Candan. *Issues and Evaluations of Caching Solutions for Web Application Acceleration*. In Proceedings of the International Conference on Very Large Data Bases (VLDB 02'), Hong Kong, China, August 2002, pages 1019–1030.
- [Li d] Y. Li, T. Lin, H. Tang & P. Sun. *A Chunk Caching Location and Searching Scheme in Content Centric Networking*. In Proceedings of the IEEE International Conference

on Communications (ICC), Ottawa, Ontario, Canada, June 2012, pages 2655–2659.

- [Li e] Z. Li & G. Simon. *Time-shifted TV in Content Centric Networks: The Case for Cooperative In-network Caching*. In Proceedings of the IEEE International Conference on Communications (ICC'11), Kyoto, Japan, June 2011, pages 1–6.
- [Lua] E.K. Lua, J. Crowcroft, M. Pias, R. Sharma & S. Lim. *A Survey and Comparison of Peer-to-peer Overlay Network Schemes*. IEEE Communications Surveys and Tutorials, April 2005, vol. 7, no. 2, pages 72–93.
- [Ma] R.T.B. Ma & D. Towsley. *Cashing in on Caching: On-demand Contract Design with Linear Pricing*. In Proceedings of the International Conference on Emerging Networking Experiments and Technologies (CoNEXT'15), Heidelberg, Germany, December 2015, pages 1–6.
- [Mahanti] A. Mahanti, D. Eager & C. Williamson. *Temporal Locality and its Impact on Web Proxy Cache Performance*. Elsevier Journal on Performance Evaluation, October 2000, vol. 42, no. 2, pages 187–203.
- [Mannes] E. Mannes & C. Maziero. *Naming Content on the Network Layer*. ACM Computing Surveys, June 2019, vol. 52, no. 3, pages 1–28.

- [Mastorakis a] S. Mastorakis, A. Afanasyev, I. Moiseenko & L. Zhang. *ndnSIM 2: An updated NDN simulator for NS-3*. Rapport technique, NDN Technical Report, NDN-0028, Revision 2, November 2016.
- [Mastorakis b] S. Mastorakis, A. Afanasyev & L. Zhang. *On the Evolution of ndnSIM: an Open-Source Simulator for NDN Experimentation*. ACM SIGCOMM Computer Communication Review, July 2017, vol. 47, no. 3, pages 19–33.
- [Mazieres] D. Mazieres, M. Kaminsky, M.F. Kaashoek & E. Witchel. *Separating key management from file system security*. In Proceedings of the ACM Symposium on Operating Systems Principles (SOSP'99), Charleston, South Carolina, USA, December 1999, pages 124–139.
- [Ming] Z. Ming, M. Xu & D. Wang. *Age-based Cooperative Caching in Information-Centric Networks*. In Proceedings of the IEEE INFOCOM Workshop on Emerging Design Choices in Named-Oriented Networking (NOMEN'12), Orlando, Florida, USA, March 2012, pages 268–273.
- [Muscariello a] L. Muscariello. *Cisco Advances Open-Source Hybrid Information-Centric Networking for 5G*, *Cisco Blogs*. (February 2019) Retrieved from <https://blogs.cisco.com/innovation/cisco-advances-open-source-hybrid-information-centric-networking-for-5g>.

- [Muscariello b] L. Muscariello, G. Carofiglio & M. Gallo. *Bandwidth and storage sharing performance in information centric networking*. In Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking (ICN'11), August 2011, Toronto, Canada, pages 26–31.
- [Oran] D. Oran. *Caching Dynamic Content*, *NDN Mailing List*. (May 2017), Retrieved from <https://www.lists.cs.ucla.edu/pipermail/ndn-interest/2017-May/001706.html>.
- [Pacifci] V. Pacifci & G. Dan. *Content-peering Dynamics of Autonomous Caches in a Content-Centric Network*. In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'13), Turin, Italy, April 2013, pages 1–9.
- [Panwar] G. Panwar, S. Misra, R. Tourani & A. Mthibaa. *Request Aggregation: The Good, The Bad, and The Ugly*. In Proceedings of the ACM Conference on Information-Centric Networking (ICN'17), Berlin, Germany, September 2017, pages 198–199.
- [Passarella] A. Passarella. *A Survey on Content-Centric Technologies for the Current Internet: CDN and P2P Solutions*. Elsevier Journal on Computer Communications, January 2011, vol. 35, no. 1, pages 1–32.
- [Pentikousis] K. Pentikousis, B. Ohlman, E. Davies, S. Spirou & G. Bo-

gia. *Information-Centric Networking: Evaluation and Security Considerations*. Rapport technique, Internet Research Task Force, RFC 7945, September 2016.

- [Perino] D. Perino & M. Varvello. *A Reality Check for Content Centric Networking*. In Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking (ICN'11), Toronto, Ontario, Canada, August 2011, pages 44–49.
- [Pham] T.M. Pham, S. Fdida & P. Antoniadis. *Pricing in Information-Centric Network Interconnection*. In Proceedings of the International Federation for Information Processing Networking Conference (IFIP'13), Brooklyn, New York, USA, May 2013, pages 1–9.
- [Project a] NDN Project. *NDN Packet Format Specification 0.3 documentation, Named Data Networking*. Retrieved February 2019 from <https://named-data.net/doc/NDN-packet-spec/current/index.html>.
- [Project b] NS-3 Project. *Events and Simulator, A Discrete-Event Network Simulator*. (June 2019), Retrieved from <https://www.nsnam.org/docs/manual/html/events.html>.
- [Project c] NS-3 Project. *RealTime, A Discrete-Event Network Simulator*. (June 2019), Retrieved from <https://www.nsnam.org/docs/manual/html/realtime.html>.

- [Psaras a] I. Psaras, W.K. Chai & G. Pavlou. *In-network Cache Management and Resource Allocation for Information-Centric Networks*. IEEE Transactions on Parallel and Distributed Systems, December 2014, vol. 25, no. 11, pages 2920–2931.
- [Psaras b] I. Psaras, W.K. Chai & G. Pavlou. *Probabilistic In-network Caching for Information-Centric Networks*. In Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking (ICN'12), Orlando, Florida, USA, March 2012, pages 55–60.
- [Rajahalme] J. Rajahalme, M. Sarela, P. Nikander & S. Tarkoma. *Incentive-compatible Caching and Peering in Data-Oriented Networks*. In Proceedings of the International Conference on Emerging Networking Experiments and Technologies (CoNEXT'08), Madrid, Spain, December 2008, pages 1–6.
- [Rossi a] D. Rossi & G. Rossini. *Caching Performance of Content Centric Networks under Multi-path Routing (and more)*. Rapport technique, Telecom ParisTech Ecole, Paris, France, November 2011.
- [Rossi b] D. Rossi & G. Rossini. *On Sizing CCN Content Stores by Exploiting Topological Information*. In Proceedings of the IEEE INFOCOM Workshop on Emerging Design Choices in Named-Oriented Networking (NOMEN'12), Orlando, Florida, USA, March 2012, pages 1–6.

- [Rossini a] G. Rossini & D. Rossi. *Coupling Caching and Forwarding: Benefits, Analysis and implementation*. In Proceedings of the ACM Conference on Information-Centric Networking (ICN'14), Paris, France, September 2014, pages 127–136.
- [Rossini b] G. Rossini & D. Rossi. *A dive into the caching performance of content centric networking*. In Proceedings of the International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD'12), Barcelona, Spain, September 2012, pages 105–109.
- [Saha] S. Saha, A. Lukyanenko & A. Yia-Jaaski. *Cooperative Caching Through Routing Control in Information-Centric Networks*. In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'13), Turin, Italy, April 2013, pages 1–9.
- [Saino a] L. Saino, I. Psaras & G. Pavlou. *Hash-routing Schemes for Information Centric Networking*. In Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking (ICN'13), Hong Kong, China, September 2013, pages 27–32.
- [Saino b] L. Saino, I. Psaras & G. Pavlou. *Icarus: a Caching Simulator for Information Centric Networking (ICN)*. In Proceedings of the EAI International Conference on Simulation Tools and Techniques (SIMUtools'14), Lisbon, Portugal, March 2014, pages 1–10.

- [Salah] H. Salah, M. Alfatafta, S. SayedAhmed & T. Strufe. *CoMon++: Preventing Cache Pollution in NDN Efficiently and Effectively*. In Proceedings of the IEEE Conference on Local Computer Networks (LCN), Singapore, Singapore, October 2017, pages 1–9.
- [Saltzer] J.H. Saltzer, D.P. Reed & D.D Clark. *End-to-end Arguments in System Design*. ACM Transactions on Computer Systems, November 1984, vol. 2, no. 4, pages 277–288.
- [Sandvine] Sandvine. *The Global Internet Phenomena Report*. Rapport technique, Sandvine Intelligent Broadband Networks, June 2013.
- [Shannigrahi] S. Shannigrahi, C. Fan & C. Papadopoulos. *Request Aggregation, Caching, and Forwarding Strategies for Improving Large Climate Data Distribution with NDN: A Case Study*. In Proceedings of the ACM Conference on Information-Centric Networking (ICN’17), Berlin, Germany, September 2017, pages 54–65.
- [Shi a] J. Shi. *Memory and Time Consumption-ndnSIM(v1) Simulations, NDN Mailing List*. (June 2019), Retrieved from <https://www.lists.cs.ucla.edu/pipermail/ndnsim/2019-June/005407.html>.
- [Shi b] J. Shi. *Memory Problem ndnSIM, NDN Mailing List*. (April 2019), Retrieved from

<https://www.lists.cs.ucla.edu/pipermail/ndnsim/2017-April/003657.html>.

- [Sourlas a] V. Sourlas, P. Flegkas, L. Gkatzikis & L. Tassiulas. *Autonomic Cache Management in Information-Centric Networks*. In Proceedings of the IEEE International Network Operations and Management Symposium (NOMS'12), Maui, Hawaii, USA, April 2012, pages 121–129.
- [Sourlas b] V. Sourlas, P. Flegkas, G.S. Paschos, D. Katsaros & L. Tassiulas. *Storage Planning and Replica Assignment in Content-Centric Publish/Subscribe Networks*. Elsevier Journal on Computer Networks, December 2011, vol. 55, no. 18, pages 4021–4032.
- [Sourlas c] V. Sourlas, P. Flegkas & L. Tassiulas. *A Novel Cache Aware Routing Scheme for Information-Centric Networks*. Elsevier Journal on Computer Networks, February 2014, vol. 59, pages 44–61.
- [Sourlas d] V. Sourlas, G.S. Paschos, P. Flegkas & L. Tassiulas. *Caching in Content-based Publish/Subscribe Systems*. In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '09), Honolulu, Hawaii, USA, November 2009, pages 1–6.
- [Spring] N. Spring, R. Mahajan & D. Wetherall. *Measuring ISP*

topologies with Rocketfuel. IEEE/ACM Transactions on Networking, February 2004, vol. 12, no. 1, pages 2–16.

- [Team 14] NDN Project Team. *NDN Technical Memo: Naming Conventions*. Rapport technique, 2014.
- [Tortelli] M. Tortelli, D. Rossi, G. Boggia & L.A. Grieco. *ICN Software Tools: Survey and Cross-comparison*. Elsevier Journal on Simulation Modelling Practice and Theory, April 2016, vol. 63, pages 23–46.
- [Tyson] G. Tyson, N. Sastry, R. Cuevas, I. Rimac & A. Mauthe. *A survey of mobility in information-centric networks*. In Proceedings of the MoBiHoc Workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms and Applications (NoM'12), Hilton Head Island, South Carolina, USA, June 2012, pages 1–6.
- [Vakali] A. Vakali & G. Pallis. *Content Delivery Networks: Status and Trends*. IEEE Journal on Internet Computing, December 2003, vol. 7, no. 6, pages 68–74.
- [Virgilio] M. Virgilio, G. Marchetto & R. Sisto. *PIT Overload Analysis in Content Centric Networks*. In Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking (ICN'13), Hong Kong, China, August 2013, pages 67–72.
- [Walfisha] M. Walfisha, H. Balakrishnana & S. Shenkerb. *Untangling the Web from DNS*. In Proceedings of the Symposium on

Networked Systems Design and Implementation (NSDI'04), San Francisco, California, USA, March 2004, pages 1–17.

- [Wang a] J.M. Wang & B. Bensaou. *Progressive Caching in CCN*. In Proceedings of the IEEE Global Communications Conference (GLOBECOM'12), Anaheim, California, USA, December 2012, pages 2727–2732.
- [Wang b] J.M. Wang, J. Zhang & B. Bensaou. *Intra-AS Cooperative Caching for Content-Centric Networks*. In Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking (ICN'13), Hong Kong, China, August 2013, pages 61–66.
- [Wang c] L. Wang, G. Tyson, J. Kangasharju & J. Crowcroft. *Fair-Cache: Introducing Fairness to ICN Caching*. In Proceedings of the IEEE International Conference on Network Protocols (ICNP'16), Singapore, Singapore, November 2016, pages 1–10.
- [Wang d] Y. Wang, N. Rozhnova, A. Narayanan, D. Oran & I. Rhee. *An Improved Hop-by-hop Interest Shaper for Congestion Control in Named Data Networking*. In Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking (ICN'13), Hong Kong, China, August 2013, pages 55–60.
- [Wang e] Y. Wang, M. Xu & Z. Feng. *Hop-based Probabilistic Caching for Information-Centric Networks*. In Proceedings

of the IEEE Global Communications Conference (GLOBECOM'13), Atlanta, Georgia, USA, December 2013, pages 2102–2107.

[Wang 14] W. Wang, Y. Sun, Y. Guo, D. Kaafar, J. Jin, J. Li & Z. Li. *CRCache: exploiting the correlation between content popularity and network topology information for ICN caching*. In Proceedings of the IEEE International Conference on Communications (ICC'14), June 2014, Sydney, Australia, pages 3191–3196, 2014.

[Wong] T.M. Wong & J. Wilkes. *My Cache Or Yours? Making Storage More Exclusive*. In Proceedings of the USENIX Annual Technical Conference (ATEC'02), Monterey, California, USA, June 2002, pages 161–175.

[Xiaoyan] H. Xiaoyan & G. Jian. *Opportunistic On-Path Caching for Named Data Networking*. IEICE Transactions on Communications, November 2014, vol. 97, no. 11, pages 2360–2367.

[Xie] M Xie, I. Widjaja & H. Wang. *Enhancing Cache Robustness for Content-Centric Networking*. In Proceedings of the IEEE International Conference on Communications (INFOCOM'12), Orlando, Florida, USA, March 2012, pages 2426–2434.

[Xu] Z. Xu, B. Chen, N. Wang, Y. Zhang & Z. Li. *ELDA: Towards Efficient and Lightweight Detection of Cache Pollution*

Attacks in NDN. In Proceedings of the IEEE Conference on Local Networks (LCN'15), Clearwater Beach, Florida, USA, October 2015, pages 82–90.

- [Xylomenos] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros & G. Polyzos. *A Survey of Information-Centric Networking Research*. IEEE Communications Surveys and Tutorials, July 2013, vol. 16, no. 2, pages 1–26.
- [Yi a] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang & L. Zhang. *A Case for Stateful Forwarding Plane*. Elsevier Journal on Computer Communications, April 2013, vol. 36, no. 7, pages 779–791.
- [Yi b] C. Yi, A. Afanasyev, L. Wang, B. Zhang & L. Zhang. *Adaptive Forwarding in Named Data Networking*. ACM SIGCOMM Computer Communication Review, July 2012, vol. 42, no. 3, pages 62–67.
- [Yu a] H. Yu, L. Xie & S. Sanner. *The Lifecycle of a YouTube Video: Phases, Content and Popularity*. In Proceedings of the International AAAI Conference on Web and Social Media (ICWSM'15), Oxford, UK, May 2015, pages 1–10.
- [Yu b] Y. Yu, A. Afanasyev, D. Clark, K. C. Claffy, V. Jacobson & L. Zhang. *Schematizing Trust in Named Data Networking*. In Proceedings of the ACM Conference on Information-

Centric Networking (ICN'15), San Fransisco, California, USA, September 2015, pages 177–186.

- [Zegura] E.W. Zegura, K.L. Calvert & S. Bhattacharjee. *How to Model an Internetwork*. In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM'96), San Francisco, California, USA, March 1996, pages 1–9.
- [Zhang a] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang & B. Zhang. *Named Data Networking*. ACM SIGCOMM Computer Communication Review, July 2014, vol. 44, no. 3, pages 66–73.
- [Zhang b] Z. Zhang, Y. Yu, H. Zhang, E. Newberry, S. Mastorakis, Y. Li, A. Afanasyev & L. Zhang. *An Overview of Security Support in Named Data Networking*. IEEE Communications Magazine, November 2017, vol. 56, no. 1–7, pages 62–68.
- [Zhou a] J. Zhou, Y. Li, V.K. Adhikari & Z.L. Zhang. *Counting Youtube Videos via Random Prefix Sampling*. In Proceedings of the Internet Measurement Conference (IMC'11), Toronto, Ontario, Canada, August 2011, pages 371–380.
- [Zhou b] J. Zhou, Q. Wu, Z. Li, M.A. Kaafar & G. Xie. *A Proactive Transport Mechanism with Explicit Congestion Notification for NDN*. In Proceedings of the IEEE International Conference on Communications (ICC'15), London, UK, June 2015, pages 5242–5247.

[Zhu]

Y. Zhu, M. Chen & A. Nakao. *Conic: Content-Oriented Network with Indexed Caching*. In Proceedings of the IEEE INFOCOM Workshops (INFOCOM'10), San Diego, California, USA, March 2010, pages 1–6.