Design and Analysis of an Information-Centric Protocol Architecture in Softwarized Mobile Networks

Original Citation:
Design and Analysis of an Information-Centric Protocol Architecture in Softwarized Mobile Networks / Benedetti, Paolo. - ELETTRONICO. - (2022). [10.60576/poliba/iris/benedetti-paolo_phd2022]

Availability:
This version is available at http://hdl.handle.net/11589/232748 since: 2021-12-28

Published version
Politecnico di Bari
DOI:10.60576/poliba/iris/benedetti-paolo_phd2022

Department of Electrical and Information Engineering
ELECTRICAL AND INFORMATION ENGINEERING
Ph.D. Program
SSD: ING-INF/03–TELECOMMUNICATIONS

Final Dissertation

Design and Analysis of an Information-Centric Protocol Architecture in Softwarized Mobile Networks

---

by

Paolo Benedetti:

Supervisor:

Prof. L. Alfredo Grieco

Coordinator of Ph.D. Program:
Prof. Mario Carpentieri

Course n°34, 01/11/2018-31/10/2021

Politecnico
di Bari

LIBERATORIA PER L'ARCHIVIAZIONE DELLA TESI DI DOTTORATO

Al Magnifico Rettore
del Politecnico di Bari

Il/la sottoscritto/a Paolo Benedetti nato/a a Fasano (BR) il 03/01/1995 residente a Fasano (BR) in via Roma, 285 e-mail paolo.benedetti95@gmail.com iscritto al 3° anno di Corso di Dottorato di Ricerca in Ing. Elettrica e dell'Informazione ciclo 34° ed essendo stato ammesso a sostenere l'esame finale con la prevista discussione della tesi dal titolo:

Design and Analysis of an Information-Centric Protocol Architecture in Softwarized Mobile Networks

DICHIARA

1) di essere consapevole che, ai sensi del D.P.R. n. 445 del 28.12.2000, le dichiarazioni mendaci, la falsità negli atti e l'uso di atti falsi sono puniti ai sensi del codice penale e delle Leggi speciali in materia, e che nel caso ricorressero dette ipotesi, decade fin dall'inizio e senza necessità di nessuna formalità dai benefici conseguenti al provvedimento emanato sulla base di tali dichiarazioni;
2) di essere iscritto al Corso di Dottorato di ricerca in Ing. Elettrica e dell'Informazione ciclo 34°, corso attivato ai sensi del "*Regolamento dei Corsi di Dottorato di ricerca del Politecnico di Bari*", emanato con D.R. n.286 del 01.07.2013;
3) di essere pienamente a conoscenza delle disposizioni contenute nel predetto Regolamento in merito alla procedura di deposito, pubblicazione e autoarchiviazione della tesi di dottorato nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica;
4) di essere consapevole che attraverso l'autoarchiviazione delle tesi nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica del Politecnico di Bari (IRIS-POLIBA), l'Ateneo archivierà e renderà consultabile in rete (nel rispetto della Policy di Ateneo di cui al D.R. 642 del 13.11.2015) il testo completo della tesi di dottorato, fatta salva la possibilità di sottoscrizione di apposite licenze per le relative condizioni di utilizzo (di cui al sito http://www.creativecommons.it/Licenze), e fatte salve, altresì, le eventuali esigenze di "embargo", legate a strette considerazioni sulla tutelabilità e sfruttamento industriale/commerciale dei contenuti della tesi, da rappresentarsi mediante compilazione e sottoscrizione del modulo in calce (Richiesta di embargo);
5) che la tesi da depositare in IRIS-POLIBA, in formato digitale (PDF/A) sarà del tutto identica a quelle **consegnate**/inviate/da inviarsi ai componenti della commissione per l'esame finale e a qualsiasi altra copia depositata presso gli Uffici del Politecnico di Bari in forma cartacea o digitale, ovvero a quella da discutere in sede di esame finale, a quella da depositare, a cura dell'Ateneo, presso le Biblioteche Nazionali Centrali di Roma e Firenze e presso tutti gli Uffici competenti per legge al momento del deposito stesso, e che di conseguenza va esclusa qualsiasi responsabilità del Politecnico di Bari per quanto riguarda eventuali errori, imprecisioni o omissioni nei contenuti della tesi;
6) che il contenuto e l'organizzazione della tesi è opera originale realizzata dal sottoscritto e non compromette in alcun modo i diritti di terzi, ivi compresi quelli relativi alla sicurezza dei dati personali; che pertanto il Politecnico di Bari ed i suoi funzionari sono in ogni caso esenti da responsabilità di qualsivoglia natura: civile, amministrativa e penale e saranno dal sottoscritto tenuti indenni da qualsiasi richiesta o rivendicazione da parte di terzi;
7) che il contenuto della tesi non infrange in alcun modo il diritto d'Autore né gli obblighi connessi alla salvaguardia di diritti morali od economici di altri autori o di altri aventi diritto, sia per testi, immagini, foto, tabelle, o altre parti di cui la tesi è composta.

Luogo e data Bari, 26/12/2021 Firma Paolo Benedetti

Il/La sottoscritto, con l'autoarchiviazione della propria tesi di dottorato nell'Archivio Istituzionale ad accesso aperto del Politecnico di Bari (POLIBA-IRIS), pur mantenendo su di essa tutti i diritti d'autore, morali ed economici, ai sensi della normativa vigente (Legge 633/1941 e ss.mm.ii.),

CONCEDE

- al Politecnico di Bari il permesso di trasferire l'opera su qualsiasi supporto e di convertirla in qualsiasi formato al fine di una corretta conservazione nel tempo. Il Politecnico di Bari garantisce che non verrà effettuata alcuna modifica al contenuto e alla struttura dell'opera.
- al Politecnico di Bari la possibilità di riprodurre l'opera in più di una copia per fini di sicurezza, back-up e conservazione.

Luogo e data Bari, 26/12/2021 Firma Paolo Benedetti

Politecnico
di Bari

Department of Electrical and Information Engineering
ELECTRICAL AND INFORMATION ENGINEERING
Ph.D. Program
SSD: ING-INF/03–TELECOMMUNICATIONS

Final Dissertation

Design and Analysis of an Information-Centric Protocol Architecture in Softwarized Mobile Networks

---

by

Paolo Benedetti:
Paolo Benedetti

Supervisor:

Prof. L. Alfredo Grieco

Referees:

Prof. Andrea Detti

Prof. Claudia Campolo

Coordinator of Ph.D Program:
Prof. Mario Carpentieri
Mario Carpentieri

Course n°34, 01/11/2018-31/10/2021

Declaration of Authorship

I, Paolo BENEDETTI, declare that this thesis titled, "Design and Analysis of an Information-Centric Protocol Architecture in Softwarized Mobile Networks" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Paolo Benedetti

Date: 28/12/2021

"Omnia, Lucili, aliena sunt, tempus tantum nostrum est; [...] interim hoc unum est quod ne gratus quidem potest reddere."

Seneca

POLITECNICO DI BARI

Abstract

Politecnico di Bari
Department of Electric and Information Engineering

Doctor of Philosophy

Design and Analysis of an Information-Centric Protocol Architecture in Softwarized Mobile Networks

by Paolo BENEDETTI

Information-Centric Networking emerged as one of the most promising technologies of the Future Internet and a powerful enabling technology for the provisioning of scalable and efficient services in mobile architectures. To serve mobile consumers, most of scientific contributions extend the information-centric communication primitives by means of a pull-based methodology, according to which the mobile consumer issues pending requests every time it reaches a new network attachment point. This approach generates two important shortcomings. First, the requests delivered before the handover will generate stale paths with wrong forwarding information in their network routers. As a consequence, some new contents will be delivered also to previous locations, thus wasting bandwidth and energy. Second, during handovers, mobile consumers may miss some contents released in real-time. In order to solve these issues, this work conceives a novel protocol architecture that properly customizes the functionalities of Information-Centric Networking, Multi-access Edge Computing, and Software Defined Networking paradigms. The impact of the devised protocol architecture on the communication overhead, router memory, and energy consumption is analytically formulated and evaluated in scenarios with different topology, mobility, application settings, and number of mobile consumers. The conducted tests demonstrate the effectiveness of the proposed protocol architecture, through computer simulations, in all the considered scenarios.

Contents

List of Figures

List of Tables

List of Acronyms

BRITE	Representative Internet Topology gEnerator
C	Consumer
CAPWAP	Control And Provisioning of Wireless Access Points
COPSS	Content-Oriented Publish/Subscribe System
CS	Content Store
FIB	Forwarding Information Base
ICN	Information-Centric Networking
ID	Identifier
IoT	Internet of Things
IP	Internet Protocol
IU	Information Unit
KPI	Key Performance Indicator
LSA	Link State Advertisement
LSDB	Link State DataBase
MAN	Metropolitan Area Network
MEC	Multi-access Edge Computing
NDN	Named-Data Networking
NG-SDN	Next Generation SDN
NLSR	Named-data Link State Routing
P	Producer
PIT	Pending Interest Table

POF	Protocol Oblivious Forwarding
PURSUIT	Publish-subscribe Internet Technology
QoS	Quality of Service
RP	Rendezvous-Point
RTT	Round Trip Time
SDC	Software-Defined Controller
SDN	Software-Defined Networking
ST	Subscription table

Personal Scientific Contributions

The most significant scientific contributions generated during the PhD are listed in what follows. Those works have been accepted for publication in international journals and conferences or they are still waiting for revision.

International Journal:

- Benedetti, P., Piro, G., & Grieco, L. A., A softwarized and MEC-enabled protocol architecture supporting consumer mobility in Information-Centric Networks. Computer Networks, 2021, 188:107867.

International Conferences:

- Benedetti, P., Ventrella, A. V., Piro, G., & Grieco, L. A., An SDN-aided information centric networking approach to publish-subscribe with mobile consumers, In 2019 Sixth International Conference on Software Defined Systems (SDS), Rome, Italy, July, IEEE, 2019, p. 130-137.
- Benedetti, P., Piro, G., & Grieco, L. A., An Energy Efficient and Software-Defined Information-Centric Networking Approach to Consumer Mobility, In 2020 22nd International Conference on Transparent Optical Networks (ICTON), Bari, Italy, IEEE, 2020, p. 1-4.
- Benedetti, P., Piro, G., & Grieco, L. A., POSTER: Analysis of the Energy Saving in Emerging Information-Centric Metropolitan Area Networks, In 2021 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), Boston, USA, IEEE, 2021, p. 1-2.

This thesis work describes the findings published in the manuscripts cited above.

The work in "An SDN-aided information centric networking approach to publish-subscribe with mobile consumers" also won the Best Paper Award at the IEEE Sixth International Conference on Software Defined Systems (SDS), held in Rome (Italy) in July 2019.

Project involvement

This thesis work was framed in the context of the Apulia Region (Italy) Research project INTENTO (36A49H6), which envisages the control and management of an optical transport network, using Software-Defined Networking and Network Function Virtualization technologies.

The project is run by Politecnico di Bari in collaboration with SM-Optics, a leading company in the software development and integration, and the optical transport sector, and Experis, a leading company in the research and selection of qualified talents, development of careers and skills, and consulting for IT, Finance, and Engineering solutions. It intends to create a software platform through which the main telecommunications companies can streamline their processes of creating advanced and energy-aware services in the fields of multimedia contents delivery, Intelligent Transportation Systems, Information Security, Video surveillance, etc.

This thesis work aims to design and analyze a novel protocol architecture capable of efficiently providing real-time services to mobile consumers in the context of the Future Internet. The reference scenario requires a dynamic and optimized configuration of network functions (e.g., the routing tables adopted by the Named-Data Networking protocol), which can be reached with the aid of sophisticated monitoring and control techniques. In this context, the conceived solution integrates Named-Data Networking, Software-Defined Networking and Multi-access Edge Computing and guarantees the achievement of the aforementioned objective, while reducing the communication overhead compared to reference solutions in literature. At the same time, the designed protocol ensures the continuity of the service and bandwidth, memory, and energy savings.

It should be noted how this high-level research activity offers evident value within the INTENTO project as it: tackles issues strictly related to the reference technologies of the network architecture being studied and developed in the INTENTO project, analyzes case studies considered by the INTENTO project and the involved companies, and offers excellent opportunities for disseminating research results in publications of high scientific profile.

Introduction

In the beginning, Internet was born to connect a limited quantity of fixed hosts to exchange crucial information on research and defense activities. But, the usage of the Internet changed considerably since its inception: nowadays massive amounts of devices with different computational capabilities and network access technologies exchange all kinds of information in mobility conditions. The issues and possibilities arising from the ever-growing complexity of the aforementioned scenario represent a primary motivation for researchers to design and analyze novel network solutions. In fact, the research activities outline the network of the future (i.e., the Future Internet) starting from the most compelling demands of current networks and the lessons learned since the inception of the Internet.

Among these novel network designs and paradigms, Information-Centric Networking (ICN) stands out for the unique advantages it can convey to network architectures, including advanced, secure, scalable, and efficient services in a wide range of network deployments. Nevertheless, the provisioning of real-time applications (such as real-time generated content from websites and live-streaming videos, or data requested by robot consumers for remote control and local decisions) to mobile consumers still represents a challenging research topic that necessitates a thorough investigation. With reference to pure ICN deployments, the majority of works published in the scientific literature assumes to manage consumer mobility through a pull-based approach: every time the consumer moves to a new network attachment point, it re-issues the set of pending requests. This basic solution, however, generates two important shortcomings, which significantly compromise the network performance and the quality of service experienced by end users. In fact, the requests delivered just before the handover leave into the network wrong forwarding information, lead to the dissemination of unuseful data to stale destinations, and inevitably generate a waste of bandwidth and energy. On the other hand, during the handover, mobile consumers may miss contents released in real-time and lose the synchronization with the remote producer. Other sophisticated approaches for consumer mobility, including tunnel-based, update-based, and locator/identifier separation schemes, are still affected by these problems and, in some cases, may magnify their negative impact on the performance (see Chapter 1 for more details).

Recent studies proposed to exploit Software-Defined Networking (SDN) and Multi-access Edge Computing (MEC) technologies to boost the potential of ICN paradigm. Thanks to its native ability to control forwarding functionalities, SDN can be adopted to manage mobility, also in ICN networks. Unfortunately, most of the available contributions focus on producer mobility and lack a thorough analysis of the impact of their proposals on the energy consumption.. Few works addressing consumer mobility propose proactive methodologies that seamlessly deliver requested contents towards the expected new positions of mobile users. Moreover, they do not solve the issues related to the presence of wrong forwarding rules and the loss of synchronization with the remote producer. At the same time, all the contributions integrating SDN facilities in ICN deployments assume to use the conventional OpenFlow protocol for network programmability purposes, despite the skepticism against this de facto standard highlighted by many research initiatives working on novel approaches, like Protocol Oblivious Forwarding (POF) and Next Generation SDN (NG-SDN). On the other hand, MEC capabilities can further improve ICN and SDN functionalities. But, also in this case, available studies do not explore the issues emerged.

To bridge this gap, the work discussed herein intends to significantly extend the current state of the art by conceiving a novel protocol architecture that successfully integrates and properly customizes the key functionalities of ICN, MEC, and SDN, solves the challenging drawbacks generated by consumer mobility, and (more in general) improves the overall network control and performance. The devised solution is very general and can be applied to a wide range of network deployments. However, by considering the recent interests of the scientific community working in this area, the discussion presented herein focuses the attention on multi-hop wireless mesh networks. Specifically, the reference scenario embraces many network attachment points randomly located within a given geographical area, providing wireless connectivity to mobile consumers, through heterogeneous communication technologies (i.e., multi-access). Each network attachment point hosts a MEC entity, which assists mobile consumers in the data retrieving process. The consumer interacts with its reference MEC entity according to the publish-subscribe mechanism. Once received a subscription request from the mobile consumer, the MEC entity starts retrieving the contents generated (in real-time) by the producer by using a request-response communication pattern. Every time the mobile consumer attaches to a network attachment point, the new reference MEC entity recovers the synchronization with the remote producer and contacts the Software-Defined Controller (SDC) for dynamically deleting wrong forwarding information still configured in intermediary nodes of the multi-hop wireless mesh network. Moreover, POF protocol is adopted to deliver control messages through information-centric communication primitives. Indeed, differently from the current state of the art, the conceived protocol architecture leverages ICN for both the data plane and the control plane.

Based on these premises, Chapter 1 reviews the current state of the art on ICN, SDN, and MEC paradigms, by posing particular attention on the contributions focusing on consumer mobility. Chapter 2 presents a preliminary solution, based on SDN, to the main drawbacks of NDN, focusing on an implementation of the publish-subscribe communication scheme. It also develops preliminary analytical models for evaluating the information exchanged by the proposed solution. Chapter 3 illustrates an advanced solution for NDN drawbacks, based on SDN and MEC, and formulates the analytical models describing the communication overhead it produces. Then, the impact of the conceived protocol architecture to the communication overhead is numerically evaluated in scenarios with different topology, mobility, and application settings, for various number of mobile consumers. Chapter 4 thoroughly analyzes the impact of the protocol architecture proposed on the energy consumption of the network, quantifying the energy saving achieved with respect to baseline pull-based approaches. Finally, the Conclusions draw the outcome of the work and summarize the future research activities.

Chapter 1

Introduction to the Future Internet

Nowadays, the usage of the Internet changes constantly, following the most recent trends and technologies. Based on this premise, efficiency and adaptability are expected to play a significant role in the network of the future (i.e., the Future Internet). In fact, Future Internet technologies will overcome the limitations of current networks and enable innovative services. In this context, ICN and SDN represent elements of paramount importance in the next generation of networks. Several works also investigated the mutual benefits of combining the two paradigms, including the compensation of existing drawbacks. This chapter analyzes the peculiarities of ICN, SDN, and other key technologies for the Future Internet.

1.1 The vision of ICN

ICN represents a revolutionary communication paradigm for the Future Internet [1]. Differently from the current Internet, it poses the attention on contents to be exchanged (i.e., information-centric logic), instead of the need to establish an end-to-end connection between communicating peers (i.e., host-centric logic). The key idea adopted by ICN is that unique names (frequently referred to as content names) identify self-secured contents and drives forwarding operations for both requests and corresponding responses. The information-centric approach is implemented in PURSUIT, DONA, Named-Data Networking (NDN), and CONET architectures [2], which generally differ for the adopted content namespace, communication scheme, and control operations. Without loss of generality, this work focuses on NDN, which represents a widely accepted architecture based on the ICN paradigm [1], [3]. However, with few adjustments, the results of this work may be extended to other architectures as well.

In NDN, the communication follows a receiver-driven and request-response scheme: the consumer requests a content by issuing an Interest packet; the network implements routing-by-name, caching, and request aggregation mechanisms and forwards back the requested content through a Data packet. An example of communication is provided in Figure 1.1.

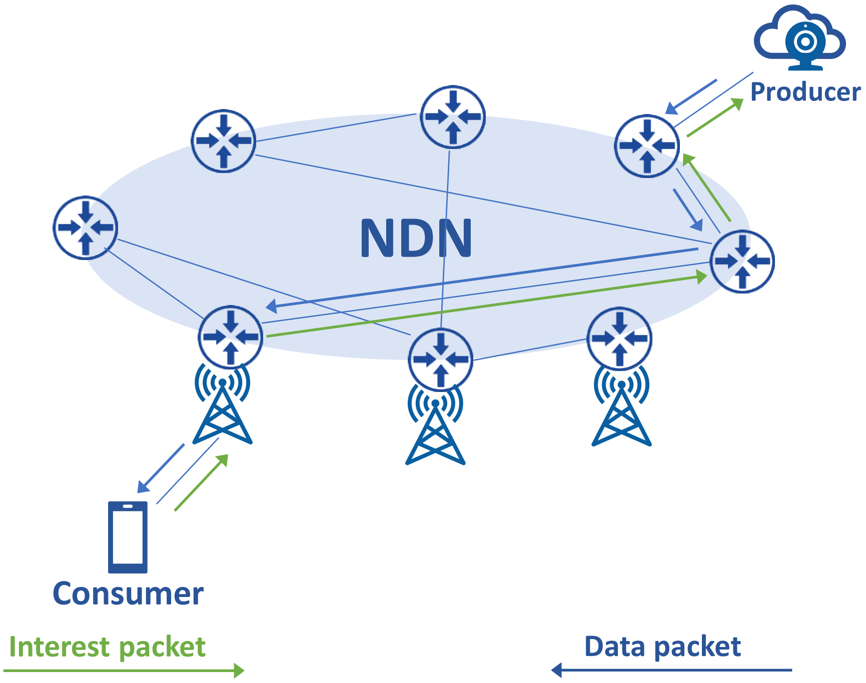


FIGURE 1.1: Example of communication in NDN.

To this end, each NDN router implements three data structures, that are: Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB). The CS of intermediate routers stores cached contents. The PIT table keeps track of received requests (i.e., Interest packets) and the network interfaces they came from. Requests for the same content are aggregated herein within a single PIT entry. This information is used to forward Data packets back to consumers across the same path of the request. Finally, the FIB table implements forwarding rules, based on the Longest Prefix Matching criteria. FIB rules are created with a routing protocol like the Named-data Link State Routing (NLSR) protocol [4], while rules in PIT tables are generated on consumer demand when an Interest packet is received.

To retrieve a given content from the network, the consumer issues an Interest packet carrying, among the other parameters, the content name. The Interest packet is delivered according to a routing-by-name algorithm, towards the first node able to satisfy the request. When a router receives an Interest packet, it may potentially perform three different tasks. First, it verifies that the requested content is already available within the local CS. If not, the router checks the presence of a similar request (i.e., a request received from another face or another consumer, but referring to the same content name) within the PIT table. If so, the router updates the retrieved PIT entry by adding the network face from which the request was received. This operation is known as Interest aggregation. Otherwise, a new entry is created and the FIB table is examined for identifying the local face through which the Interest packet has to be forwarded. The network entity able to satisfy the request (i.e., the remote producer or an internal router with a cached content) generates a Data packet, which is sent back to the consumer according to information stored within PIT tables of network routers.

NDN can also be extended to enable publish-subscribe communication strategies [5]–[9]. Available solutions are based on pull-based and push-based mechanisms. In the pull-based approach, a sliding window mechanism controls the delivery of consumer's requests. The consumer initially sends a window of Interest packets to retrieve a certain number of consecutive content updates belonging to a given topic-name. Then, every time a new Data packet is received, the sliding window moves forward and a new Interest packet is released. Instead, the pull-based approach assumes that the consumer creates a semi-permanent communication channel with the producer, through which content updates are delivered without further solicitations. In both cases, however, mobility is not natively supported.

The works in [10] and [11] propose the Content-Oriented Publish/Subscribe System (COPSS) architecture, offering publish-subscribe functionalities also in mobile conditions. Nevertheless, COPSS drastically extends NDN communication primitives, by introducing two new messages (i.e. subscribe and publish), a new logical node (i.e. Rendezvous-Point (RP)), and a new table in intermediate routers (i.e. Subscription table (ST)). More conservative mechanisms, instead, are discussed in [12], where baseline pull-based and push-based approaches are extended for supporting the mobility of both consumers and producers. In this case, the key rationale is that the consumer issues a new window of Interest packets (for the pull-based case) or a new semi-permanent Interest packet (for the push-based case) every timeout set according to the producer mobility, or every time it changes network attachment point (please, see [12] for further technical details).

1.2 The rise of Softwarization Technologies and Edge Computing

The SDN paradigm introduces a number of technical strategies for optimizing control operations and answering, in real-time, to network demands. It enables the dynamic configuration of the network through a sharp separation between control plane and data plane, thus removing control functionalities from network devices, that now simply forward packets [13]. In this context, the controller interacts with network elements through standardized APIs, namely Southbound Interface, and OpenFlow is the most popular solution supported by the Open Networking Foundation [13]. Figure 1.2 depicts a simplified and layered view of an SDN architecture.

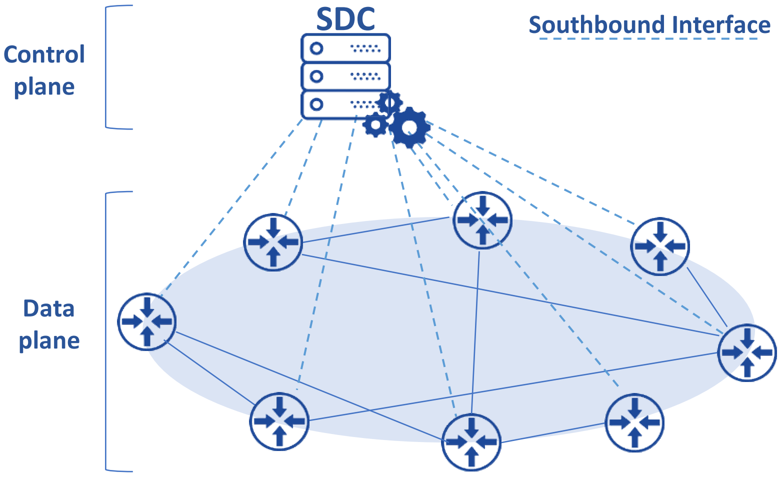


FIGURE 1.2: Simplified and layered view of an SDN architecture.

It can also slice network resources and run distinct virtual machines on programmable nodes by means of Network Hypervisors. This way, the SDC may deploy any desired network function (e.g., firewall, Dynamic Host Configuration Protocol server, Network Attached Storage) through virtualization on any node.

In general, SDN manages mobility through reactive or proactive approaches. The former, investigated in [14]–[17], supposes that the network attachment points notify the SDC every time a mobile user attaches or detaches from the network. This way, the SDC is able to update routing tables based on the knowledge of consumer mobility. The latter uses machine learning techniques to predict the position of mobile consumers and updates the network configuration accordingly. It achieves better performance in terms of delay and packet loss, but it may cause unnecessary data transmissions in case of wrong predictions [16].

Many works in the literature encompass the joint deployment of SDN and NDN with outstanding results [18]–[24]. In fact, the integration of SDN and ICN offers many advantages when compared against a "plain distributed ICN design", including caching optimization [25], [26], routing optimization [18], [23], traffic engineering [27], [28], energy efficiency [29], [30], enhanced security [24], [31], and cooperative computing management [21].

Furthermore, it is important to remark that SDN requires a flexible implementation of the control plane [32], [33]. Despite OpenFlow is still considered the de facto standard, recent novel methodologies would encourage both research and development activities to move away from OpenFlow [17], [34]–[39]. Specifically, solutions like POF [35]–[37], P4/XDP [17], [34], and Control And Provisioning of Wireless Access Points (CAPWAP) [38], [39] have been introduced to improve the programmability of the network by increasing the expressiveness of the control plane, thus enabling a faster adoption of innovative protocols. But, all the reviewed contributions tackling mobility in NDN still consider the conventional OpenFlow protocol as a Southbound Interface and the possibility to adapt different approaches seems to be unexplored, yet.

As a final remark, MEC is an emerging paradigm which brings computing power and caching to the edge of the network [40]–[43]. Network operators dispose MEC entities on the field to offer advanced services and support user mobility across multiple radio access technologies [40], while dynamically relocating context information of mobile users at the edge of the network [44], balancing network loads [45], predicting user mobility and optimizing handover [46]. Figure 1.3 shows an example of MEC architecture providing network access and edge computing to a wireless mesh network.

Some recent works propose to use the MEC entity as a gateway, aggregating requests from mobile consumers and translating these requests to NDN communication primitives [47], [48]. Moreover, combining edge caching and Interest aggregation at the MEC level considerably reduces the load in the core network [49]. The current state of the art also presents several works encompassing the integration of MEC and SDN [45], [50]–[55]. The two paradigms offer mutual benefits to each other: SDN provides centralized control to MEC and offers common APIs for network programmability in heterogeneous technologies environment, while the deployment of MEC entities improves efficiency and reliability of SDN centralized control [42].

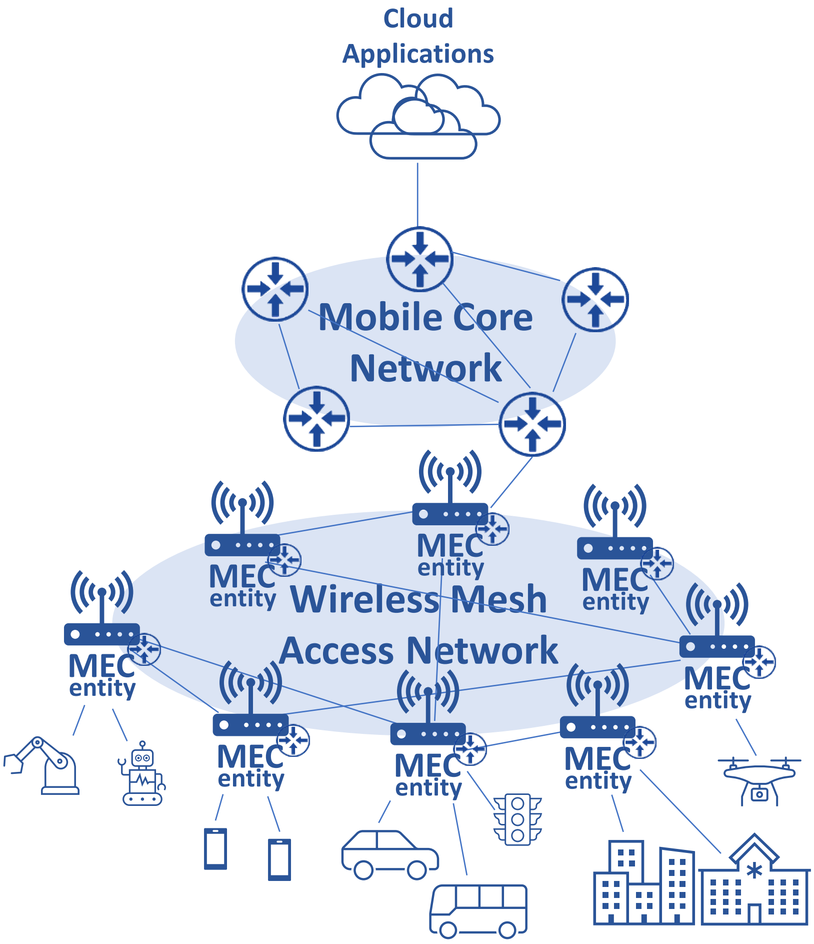


FIGURE 1.3: Example of MEC architecture in a wireless mesh network.

First attempts in the joint deployment of MEC, NDN, and SDN have been investigated in [29], [56]–[59]. The proposals in [56], [58], and [57] evaluate the possibility to orchestrate named services in edge and fog domains by exploiting the centralized control of an SDC and the adaptive forwarding plane of NDN. Other works, such as [29] and [59], propose a framework for the optimal management of network resources to improve the energy efficiency and the hit ratio, respectively. While they present an interesting perspective on architecture integration and shed a light on the promising capabilities of the framework, the contributions in [29], [56]–[59] do not propose any solution to the main drawbacks of NDN.

1.3 The drawbacks of NDN due to mobility

NDN often handles the mobility of consumers by means of a pull-based approach [12], [57], [60], [61]. A mobile consumer initially establishes a multi-hop communication path with the remote producer and requests the contents of its interest. When it moves to a new network attachment point, the consumer sends again the pending Interest packets from its updated location. Despite this baseline solution appears simple and effective [2], it generates two important drawbacks [12]. First, every time a consumer changes its position, a multi-hop communication path is created between the new attachment point and the producer. The path established before the handover, namely stale path, remains active until the related PIT entries expire. Moreover, it may remain active also after the handover process, or at the end of a long period of network detachment experienced by mobile consumers in the case of non-overlapping cells. As a consequence, contents requested before the handover could be forwarded to the old location due to wrong information stored in stale disjoint routers (that are routers of the stale path not overlapped with the new path established between consumer and producer after the handover), thus bringing to an unexpected waste of bandwidth and energy (see Figure 1.4) [12].

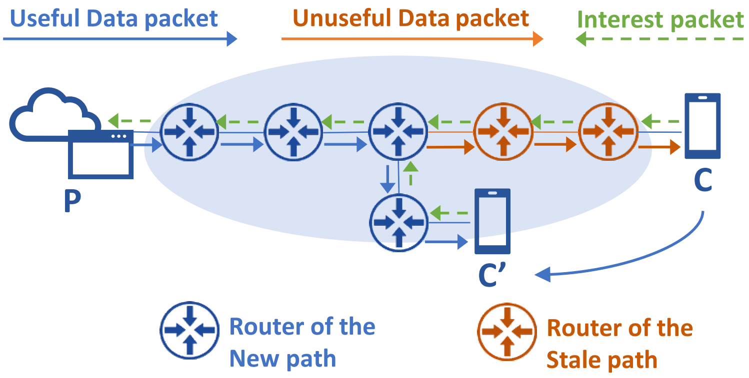


FIGURE 1.4: Example of stale path due to consumer mobility in NDN.

Second, during the handover, the consumer is unable to receive content updates. In addition, after the handover, it is also unaware of what are the latest contents available. Consequently, it is not capable to resume the data exchange with the remote producer for real-time contents (i.e., the synchronization between consumer and producer is lost) [62]. Of course, the negative impact of this issue increases in scenarios with non-overlapping cells, where the mobile consumer may experience a long period of network detachment.

More sophisticated approaches for consumer mobility are: tunnel-based, update-based, and locator/identifier separation [63]–[65]. The tunnel-based scheme leverages the triangular routing, as implemented by Mobile IP [66], [67]. This means that, when a mobile consumer moves to a new access network, it sends a binding update to its previous attachment point. Then, all packets reaching the old position are encapsulated and routed towards the new one. Optionally, anchor points can be created on demand during the handover to act as a proxy between the mobile consumer and the network [68], [69]. Update-based approaches intend to dynamically modify PIT and FIB tables according to the consumers' mobility. These updates may be triggered and/or implemented by mobile consumers, network nodes, or resource handlers [70]. Finally, the locator/identifier separation-based approach assigns a locator to each router and supposes to explain, directly into the content name, the locators of routers through which forward the data. This way, packets are forwarded across locators, without a priori knowledge of the consumer position [71]. This approach also enables proactive caching mechanisms [72]–[76]. However, proactive and tunnel-based approaches still present some problems. On the one hand, proactive approaches may solicit the forwarding of the requested contents to all nearby locations, thus producing a higher number of stale paths. Likewise, if contents are proactively forwarded to a single location (i.e., following a prediction algorithm), a new stale path is created if the prediction is wrong. On the other hand, tunnel-based approaches generate longer stale paths if mobile consumers are no longer interested in real-time contents previously requested. Furthermore, update-based approaches require extensions to the format of NDN messages or tables, lack a network-wide view, and often result in sub-optimal performance [77]. Last, none of the considered works includes a mechanism to recover the synchronization between consumer and producer after the handover.

In the same way, the pull-based publish-subscribe mechanism presented in [12] suffers from the similar drawbacks. In this case, any new version of the requested contents will also be delivered across the stale disjoint path, till its expiration. Again, the presence of wrong forwarding information in NDN routers brings to an inevitable growth of the communication overhead. Moreover, during the handover, the consumer misses some content updates and it is not able to correctly set the parameters of the sliding window mechanism. Therefore, when a new network access point is reached, the consumer should restore the synchronization with the producer, while optionally retrieving missed contents.

At the same time, the solutions presented in [60]–[62], [78]–[81] leverage SDN functionalities for mobility management in NDN networks: mobile users notify their attachment and detachment procedures to a centralized SDC and the SDC updates PIT and/or FIB tables in network nodes accordingly. While the contributions [60], [61], [80], [81] focus on VANET scenarios (e.g. vehicle speed, information on the destination, and traffic conditions) to predict the availability of producers and consumers, those discussed in [62], [78], [79] refer to broader scenarios and consider producer mobility only. Nevertheless, they do not solve the problems related to the presence of stale disjoint links and the loss of synchronization caused by the handover.

At the end of this Section, it is important to remark that this work does not propose "yet another approach" to manage seamless handover. Instead, it formulates a novel and effective protocol architecture (where the SDN controller covers a key role) that solves the issues affecting most of the strategies addressing data delivery to mobile consumers in NDN-based networks.

Table 1.1 sketches the reviewed state of the art and compares the research contributions published so far with respect to the different technical aspects investigated in this work. It is clear that different important contributions in the current literature address consumer mobility and solve the service disruption issue. But, it also emerges that this work proposes a novel protocol architecture that solves, for the first time, mobility issues related to stale paths and loss of synchronization of real-time services. It also proposes, for the first time, an information-centric implementation of the SDN control plane, based on the usage of POF capabilities. Finally, differently from several analyses available in the current scientific literature, it analytically formulates and numerically evaluates the communication overhead in scenarios with one or more mobile consumers.

At the same time, it is important to note that the methodology formulated in this work can be properly adopted to improve the behavior of any other SDN-based protocol architecture willing to address consumer mobility in an NDN deployment, like [60]–[62], [78]–[81].

TABLE 1.1: Overview on the reviewed state of the art.

Works	**NDN**	**SDN**	**Control plane implementation**	**MEC**	**Address consumer mobility**	**Solve consumer service disruption**	**Solve stale path**	**Solve loss of synchronization**	**Information-centric control plane**	**Single consumer overhead analysis**	**Multiple consumer overhead analysis**
[38], [39]		X	CAPWAP								
[35]–[37]		X	POF								
[25], [78]	X	X									
[18]–[20], [22]–[24], [26]–[28], [30]	X	X	OpenFlow								
[40]–[43], [46]				X							
[47]–[49]	X			X							
[54]		X		X							
[32], [50], [51], [53], [82]		X	OpenFlow	X							
[31]	X	X			X						
[56]	X	X		X	X						
[29], [57]–[59]	X	X	OpenFlow	X	X						
[65], [67]–[69], [71]–[74], [83], [84]	X				X	X					
[60], [61], [80], [81]	X	X			X	X					
[14]–[16]		X	OpenFlow		X	X					
[17]		X	P4		X	X					
[45]				X	X	X					
[79]	X	X								X	
[62]	X	X	OpenFlow							X	
[12], [76]	X				X					X	
[44]				X	X					X	
[66], [70], [75]	X				X	X				X	
[52], [55]		X	OpenFlow	X	X					X	
This work	**X**	**X**	**POF**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**

Chapter 2

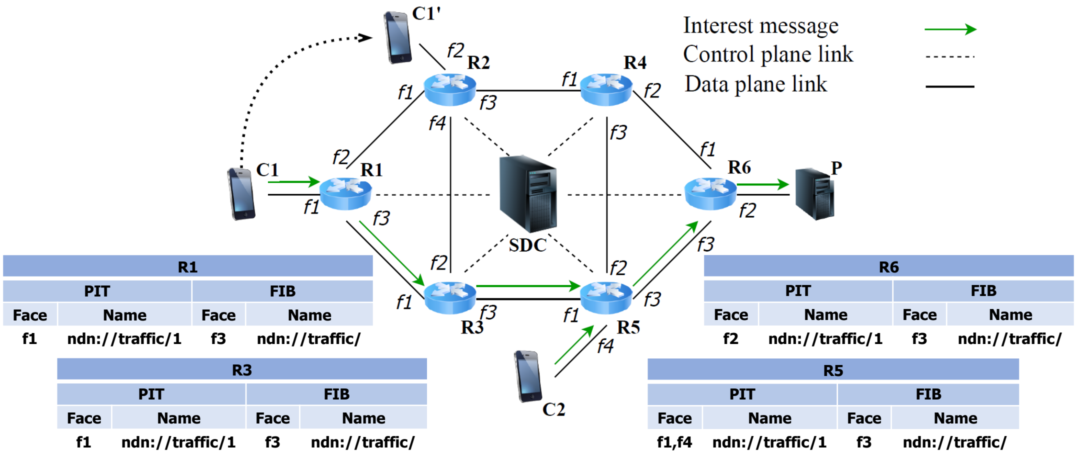
A preliminary solution to the main drawbacks of NDN

NDN represents a fundamental technology for the Future Internet. Its baseline functionalities can be extended to support the publish-subscribe communication schemes. However, in case of consumer mobility, its benefits slam against two main drawbacks. On one hand, available handover management solutions temporarily leave wrong forwarding information within network routers and do not take care of data dissemination across stale paths. On the other hand, mobile consumers inevitably lose content updates during the handover and definitively become unaware about the latest version of the content to request.

By leveraging the potentials of the Software-Defined Network paradigm, this chapter formulates new methodologies willing to solve these problems. Specifically, it proposes protocols for (1) dynamically updating forwarding functionalities through the control plane when the consumer detaches from the network and (2) restoring the synchronization between consumer and producer when the former one attaches to a new network attachment point. It addition, it develops preliminary analytical models for evaluating the average number of control information these protocols require to exchange per unit of time. The resulting analysis illustrates the pros and cons they achieve in different network configurations.

2.1 The proposed protocol architecture

Figure 2.1 shows the reference scenario investigated in this work. It embraces NDN routers, a centralized SDC, a producer, and mobile consumers. NDN routers are access points offering wireless connectivity to both consumers and producers. Connected to each other, they make a data-centric overlay network [85]. A centralized SDC interacts with all the NDN routers through the control plane. Thus, it is assumed that it knows the topology of the data-centric overlay network, the available producers, and the position of mobile consumers. For simplicity, it is assumed that a routing protocol, like the one proposed in [86], fills FIB tables of NDN routers on demand.



R1			
PIT		FIB	
Face	Name	Face	Name
f1	ndn://traffic/1	f3	ndn://traffic/

R3			
PIT		FIB	
Face	Name	Face	Name
f1	ndn://traffic/1	f3	ndn://traffic/

R6			
PIT		FIB	
Face	Name	Face	Name
f2	ndn://traffic/1	f3	ndn://traffic/

R5			
PIT		FIB	
Face	Name	Face	Name
f1,f4	ndn://traffic/1	f3	ndn://traffic/

FIGURE 2.1: Reference scenario.

The example reported in Figure 2.1 considers one producer, P, and two consumers, C_1 and C_2. The producer P generates contents under a specific topic-name, each one uniquely identified with an incremental identifier. Without loss of generality, the resulting content name is set to *ndn://[topic-name]/#id*. According to the pull-based publish-subscribe strategy [12], the number of Interest packets managed by the sliding-window mechanism is set to W. Let's now suppose that the consumer C_1 moves from R_1 to R_2. The handover process embraces detachment and attachment procedures. To solve publish-subscribe drawbacks already discussed before, these procedures are addressed differently.

2.1.1 Implementation of the detachment process

The network access point involved in the detachment process (i.e., the node R_1 in the example presented in Figure 2.1), sends a *handover initiation message* to the SDC. This message stores the consumer's identifier and the pending W requested content names. In turn, SDC updates its topology knowledge and starts updating forwarding information within the NDN network. This latter task is performed according to two possible protocols, namely *FIB inspection* and *neighbor inspection*.

FIB inspection protocol

The main rationale of the FIB inspection protocol is shown in Figure 2.2. It assumes that the SDC acquires the path established between consumer and producer before the detachment, inspects FIB and PIT tables of related NDN routers, and updates (if needed) their PIT tables by deleting wrong forwarding information associated to stale paths.

The inspection starts from the network access point that triggered the detachment processes and potentially ends when it reaches the network access point which the producer is attached to. In every round, the SDC focuses on a given router of the path and retrieves entries from its PIT and FIB tables corresponding to content names stored within the handover initiation message. To this end, the SDC and router exchange *FIB inspection request* and *FIB inspection response* messages. The SDC immediately investigates the list of faces stored within each single PIT entry. If the investigated router is the first of the path (i.e., the one that triggered the process), the SDC deletes the face associated with the mobile consumer. Otherwise, it deletes the face (if it exists) that connects the investigated router with the previous one of the path. The updated PIT entries are delivered to the router through the *FIB inspection update* message. Among the new set of PIT entries, the presence of an entry with at least one other face demonstrates that the corresponding content has been requested by another consumer, connected to the router through another path. In this case, the protocol stops looking for the corresponding content name. This way, selected stale paths previously created by the detached consumer are definitively erased. But the links belonging to other paths established with other consumers requesting the same contents through other logical faces are kept (preserving the Interest aggregation mechanism natively allowed by NDN). On the contrary, if there is an updated PIT entry with no faces, the SDC extracts from the FIB entries the neighbors through which the investigated router could forward Interest packets issued by the consumer and moves inspecting their PIT and FIB tables, as just explained.

To provide further insight, Figure 2.1 is used to discuss a practical example. The consumer C_1 issues an Interest packet for the content *ndn://traffic/1*(the example just assumes that W=1). Then, it moves before receiving the corresponding Data packet. According to the envisaged approach, R_1 sends a handover initiation message to the SDC. Now, SDC retrieves the entries related to the *ndn://traffic/1* content from PIT and FIB tables of R_1. The iterative process realizes that C_1 is the only consumer requesting the considered content through R_1. Therefore, SDC deletes the related PIT

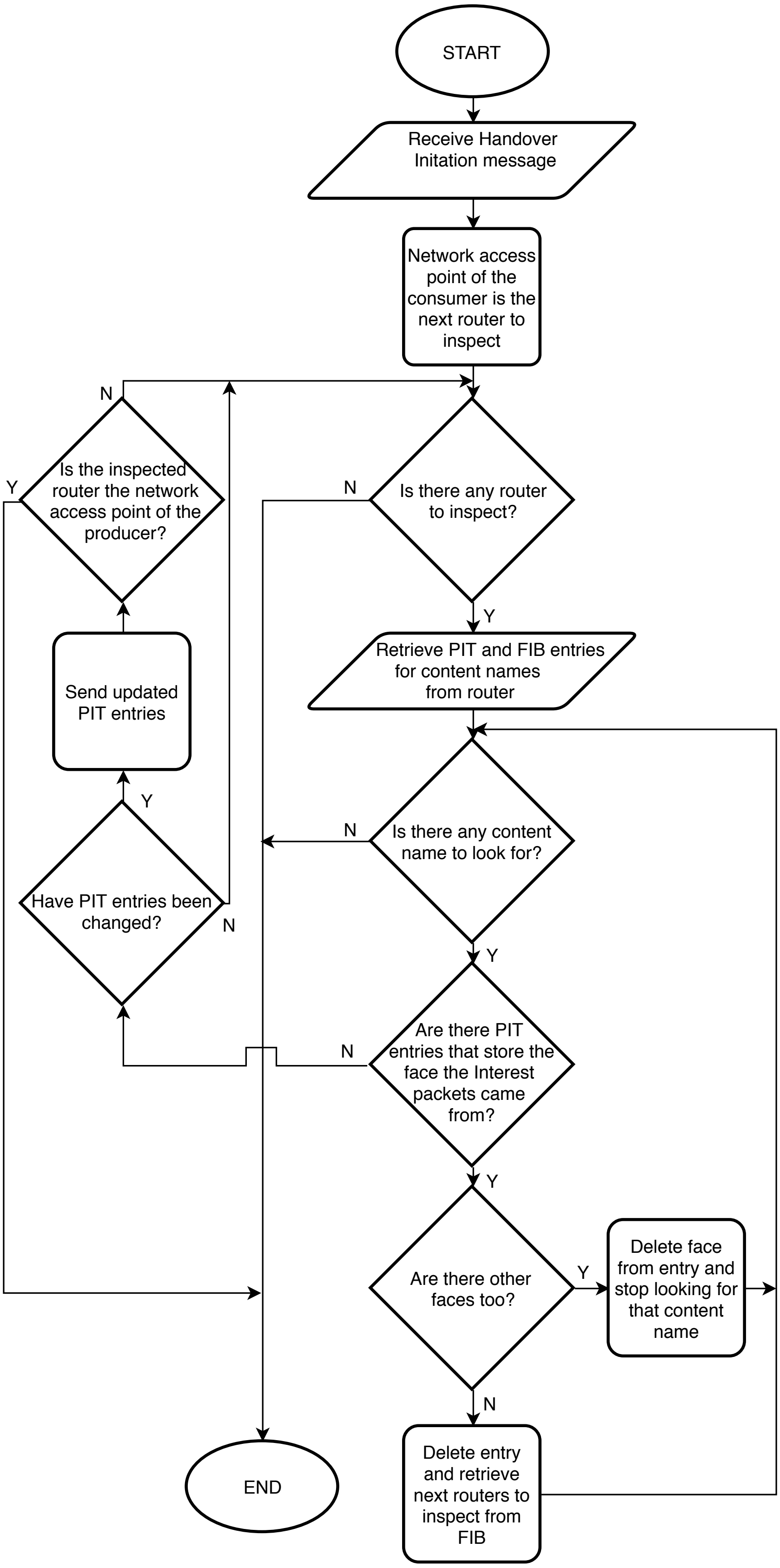


FIGURE 2.2: FIB inspection protocol

entry in R_1, inspects the FIB entry, and moves to the next NDN router of the path, that is R_3. Similarly to what happened for R_1, also in this case SDC deletes a PIT entry in R_3 and moves inspecting R_5. Here, it recognizes that the content is also requested by C_2. Therefore, the protocol updates the PIT entry of R_5 and stops running. As a result, Data packets related to the *ndn://traffic/1* content will not be sent, through a stale path, to the location left by the consumer C_1 because of the absence of breadcrumbs in PITs. Nevertheless, Data packets will still be delivered to C_2, whose path with the producer has not been modified.

Neighbor inspection protocol

The main rationale of the neighbor inspection protocol is shown in Figure 2.3. Also in this case, the inspection starts from the network access point that triggered the detachment processes and potentially ends when the network access point which the producer is attached to is reached. But, unlike the previous case, the neighbor inspection protocol intends to scavenge forwarding information stored within all the neighbors of routers belonging to the communication path established between consumer and producer before the handover. Indeed, the information stored within the FIB table is not retrieved and used.

In every round, the SDC focuses on a given router of the path and retrieves entries from its PIT tables that correspond to content names stored within the handover initiation message. To this end, SDC and router exchange *neighbor inspection request* and *neighbor inspection response* messages. In line with the FIB inspection protocol, also in this case the SDC investigates the list of faces stored within each single PIT entry. If the investigated router is the first of the path (i.e., the one that triggered the process), the SDC deletes the face associated with the mobile consumer. Otherwise, it deletes the face (if it exists) that connects the investigated router with the previous one of the path. The updated PIT entries are delivered to the router through the *neighbor inspection update* message. Among the new set of PIT entries, the presence of a PIT entry with at least one other face demonstrates that the corresponding content has been requested by another consumer, connected to the router through another path. In this case, the protocol stops looking for that content name for the same reasons discussed before. On the contrary, if there is an updated PIT entry whose face field is empty, the SDC identifies all the neighbors of the investigated router (i.e., through which it could forward Interest packets issued by the consumer) and moves to inspect their PIT tables, as just explained. To avoid redundant analysis, the router just inspected is not considered in the next rounds.

For the sake of clarity, an example is commented through Figure 2.1. The consumer C_1 issues an Interest packet for the content *ndn://traffic/1* and moves before receiving the corresponding Data packet. R_1 sends the handover initiation message to SDC. Then, SDC retrieves the entries related to the *ndn://traffic/1* content from PIT table of R_1. The iterative process realizes that C_1 is the only consumer requesting the considered content through R_1. Therefore, SDC deletes the PIT entry in R_1 and moves inspecting PIT tables of its neighbors (that are R_2 and R_3). In the considered scenario, R_2 does not have PIT entry associated with the content name of interest. Therefore, the protocol does not produce any change to its table. On the contrary, the SDC modifies the PIT entry of R_3 as well and moves inspecting its neighbors, R_5 and R_2. Here, the SDC updates the PIT entry of R_5 and the protocols stops running because a PIT entry with multiple faces is found. In the end, Data packets related to the *ndn://traffic/1* content will not be sent to the location left by the consumer C_1 because of the absence of corresponding PIT entries. Nevertheless, Data packets will still be delivered to C_2, whose path with the producer has not been modified.

2.1.2 Implementation of the attachment process

The consumer triggers the attachment process when it attaches to a new network access point. Specifically, it sends a *handover completed message*, storing its identifier and the topic-name of its interest. The new network access point forwards such a message to the SDC controller, which in turns updates the network topology details in its possession. Then, the mobile consumer retrieves the identifier of the latest content generated by the producer during the handover. With this information, the consumer restores the synchronization with the producer, which (optionally) allows it to request missed contents and start retrieving new updates. To reach this goal, two protocols are conceived, that are *SDN-based re-sync* and *NDN-based re-sync*.

SDN-based re-sync protocol

The synchronization is fully managed by the SDC, as depicted in Figure 2.4. Once the SDC receives the handover completed message, it contacts the producer and retrieves the identifier of the latest generated content. Then, it sends such information back to the consumer. To this end, new messages are

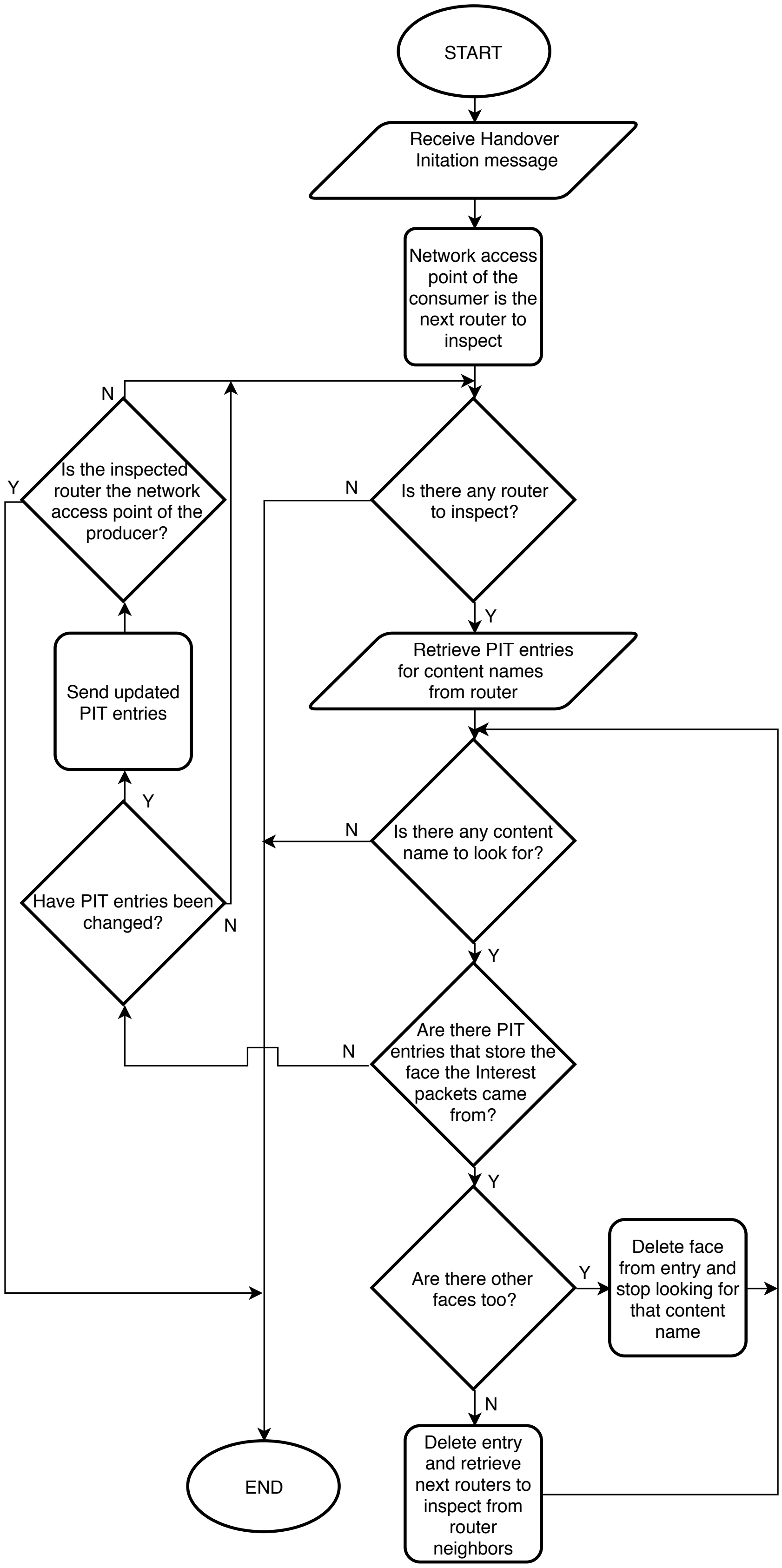


FIGURE 2.3: Neighbor inspection protocol

exchanged through the control plane: the SDC issues the *last content name request*, the producer answers with the *last content name response*, and the SDC forwards the retrieved detail to the consumer through the *last content name* message. The mobile consumer is now able to retrieve the latest content generated by the producer, the set of missed ones (if needed), and future updates. Indeed, with reference to the sliding-window mechanism, the consumer issues a window of Interest packets, whose content names are properly set based on its preferences.

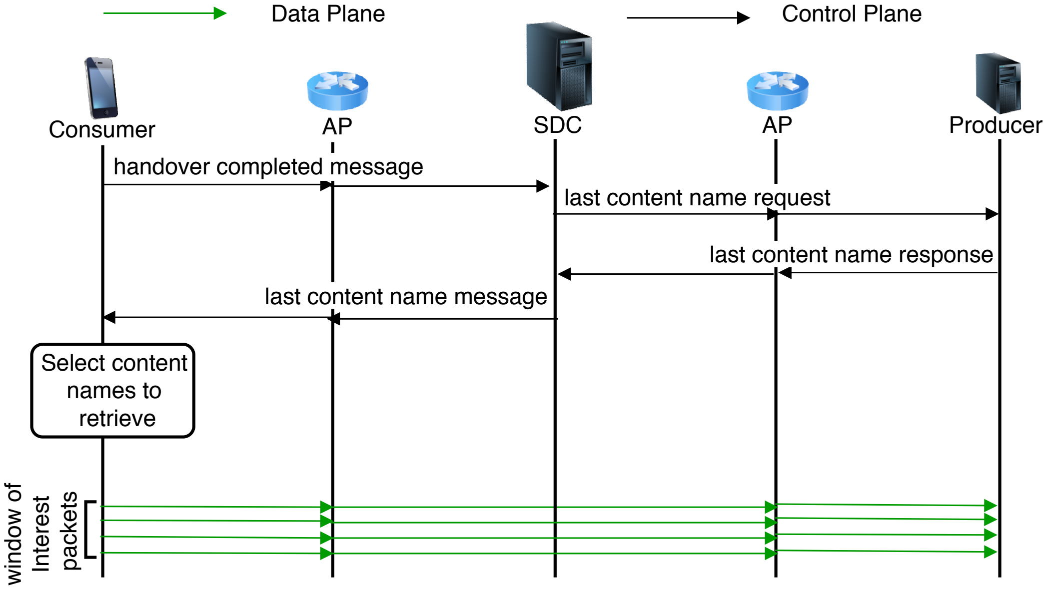


FIGURE 2.4: SDN-based re-sync protocol

NDN-based re-sync protocol

The synchronization is managed by the consumer, through NDN communication primitives. Unlike the previous case, this strategy only uses the conventional NDN data plane (see Figure 2.5). Specifically, the consumer issues an Interest packet with a special content name set to *ndn://[topic-name]/LAST/[timestamp]*. It is important to note that the field *[timestamp]* in the name is appended to avoid retrieving cached (and, hence, not updated) responses. The request is forwarded to the producer according to the NDN forwarding mechanism. The producer generates and sends back to the consumer a Data packet containing the identifier of the latest generated content. Also in this case, the mobile consumer is now able to retrieve the latest content generated by the producer, the set of missed ones (if needed), and the future updates. Therefore, a window of Interest packets is released accordingly.

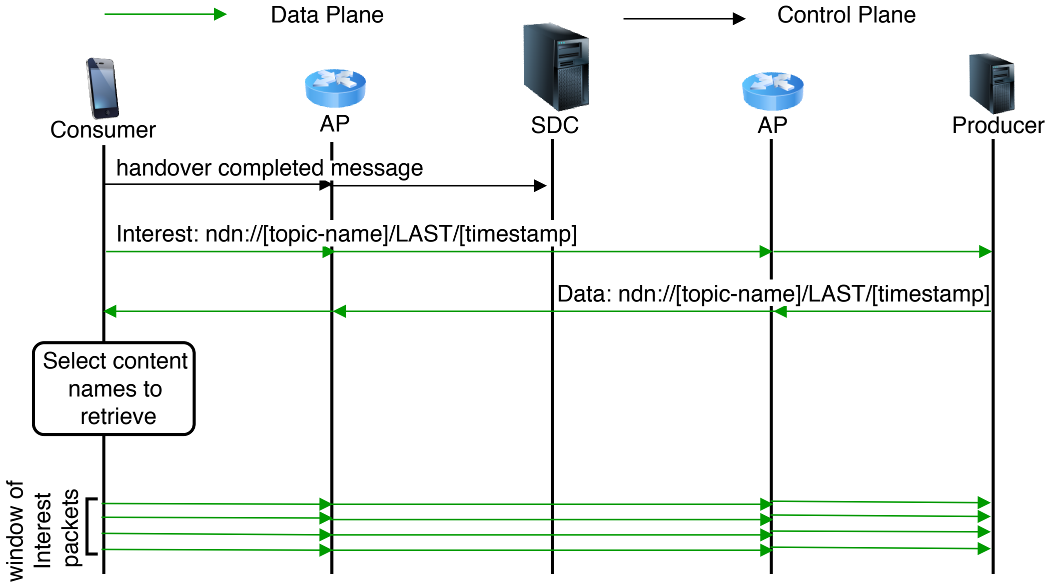


FIGURE 2.5: NDN-based re-sync protocol

2.2 Performance assessment

All the protocols described in Section 2.1 require the exchange of control information among network elements through control and data planes. Let Information Unit (IU) be the atomic detail carried by a generic control message. Considering that examples of IUs are content names, topic-names, PIT entries, and FIB entries, the number of IUs carried by each message is summarized in Table 2.1.

This section presents analytical models able to evaluate the upper bound of the average number of IUs exchanged among all the network elements in a unit of time, simply referred to as *control overhead*. The terms "upper bound" mean that the models suppose to implement each algorithm in any router of the path established, before the detachment process, between consumer and producer. To simplify the analysis, it is assumed that all the considered IUs have the same weight in the conducted study.

The control overhead also depends on the length of the path connecting any network element pair. The proposed study considers a scale-free network topology with N routers. According to [87], the number of neighbors

Protocol	Message	Type of IU	Symbol and # of carried IUs
FIB inspection	handover initiation message	content names	$H_I = W$
	FIB inspection request	content names	$F_{req} = W$
	FIB inspection response	PIT entries + FIB entry	$F_{res} = W + 1$
	FIB inspection update	PIT entries	$F_{up} = W$
Neighbor inspection	handover initiation message	content names	$H_I = W$
	neighbor inspection request	content names	$N_{req} = W$
	neighbor inspection response	PIT entries	$N_{res} = W$
	neighbor inspection update	PIT entries	$N_{up} = W$
SDN-based re-sync	handover completed message	[topic-name]	$H_C = 1$
	last content name request	content name	$L_{SDN,req} = 1$
	last content name response	content name	$L_{SDN,res} = 1$
	last content name message	content name	$L_{SDN,msg} = 1$
NDN-based re-sync	handover completed message	[topic-name]	$H_C = 1$
	Interest packet for *ndn://[topic-name]/LAST/[timestamp]*	special content name	$L_{NDN,req} = 1$
	Data packet for *ndn://[topic-name]/LAST/[timestamp]*	special content name	$L_{NDN,res} = 1$

TABLE 2.1: Number of IUs carried by each message of the proposed protocols

of each NDN router, technically named *node degree*, follows a power law distribution with the exponent parameter γ set to 3. Let k_{min}, k_{max}, and $\bar{k}$ be the minimum, the maximum, and the average node degree, respectively. For simplicity, $k_{min} = 1$. In line with [87], $k_{max} = k_{min} N^{\frac{1}{\gamma-1}}$ and the average node degree is given by $\bar{k} = \left(\sum_{k=k_{min}}^{k_{max}} k^{1-\gamma}\right) / \left(\sum_{k=k_{min}}^{k_{max}} k^{-\gamma}\right)$. The average shortest path length, i.e., $\bar{d}$, is equal to $\bar{d} = log(N)/log(log(N))$ [87], [88].

The communication path established between the consumer and the producer before the detachment process embraces the following NDN routers: $\mathcal{D} = \{r_0, r_1, r_2, ...r_{d_{old}}\}$. It embraces a number of links equal to d_{old}. The new path established after the attachment process, instead, has a number of links equal to d_{new}. The number of faces stored within the FIB entry associated with the topic-name of interest for the $i-$th router is equal to ϵ_i.

In what follows, let $d_{c,sdc}$, $d_{p,sdc}$, $d_{c,p}$, and $d_{r_{ij},sdc}$ be the shortest path established between the access point which the consumer is attached to and the SDC, the access point which the producer is attached to and the SDC, the network attachment points of both consumer and producer, and the j-th neighbor of the i-th router of $\mathcal{D}$ and the SDC, respectively. As expected, all of these paths have the same average value, that is: $E[d_{c,sdc}] = E[d_{p,sdc}] = E[d_{c,p}], = E[d_{r_{ij},sdc}] = \bar{d}$.

In line with [12], the cell residence time Δt is defined as the amount of time in which the consumer remains connected to a given network attachment point. Its average value, that is $E[\Delta t]$, can be calculated by considering the

TABLE 2.2: List of adopted symbols.

Symbol	Description
Δt	Cell residence time of the Consumer
N	Number of routers in the network topology
v	Average consumer speed
r	Average cell coverage radius
γ	Scale-free network power-law parameter
$\mathcal{D}$	Routers belonging to communication path between the consumer and the producer before the detachment
$d_{c,sdc}$	Shortest path length between the Consumer and the SDC
$d_{p,sdc}$	Shortest path length between the Producer and the SDC
$d_{c,p}$	Shortest path length between the network attachment points of the Consumer and the Producer
$d_{rij,sdc}$	Shortest path length between the j-th neighbor of the i-th router of $\mathcal{D}$ and the SDC
d_{old}	Number of links between the consumer and the producer before the detachment
d_{new}	Number of links between the consumer and the producer after the attachment
$\bar{d}$	Average shortest path length between two network nodes
k_{max}	Maximum number of neighbors per router
k_{min}	Minimum number of neighbors per router
$\bar{k}$	Average number of neighbors per router
ϵ_i	Number of faces stored within the FIB entry associated with the topic-name of interest of the i-th router
$\bar{\epsilon}$	Average number of faces stored within a FIB entry
W	Sliding window size for the pull-based approach

average radius of coverage area of network access points, r, and the average consumer speed, v. Thus, it holds that $E[\Delta t] = \pi r/2v$, as already shown in [12].

The control overhead due to a specific protocol is calculated as the ratio between the total number of exchanged IUs and the residence time, as described below. The former contribution embraces the IUs exchanged hop-by-hop on all the involved links.

A summary of the main symbols adopted is presented in Table 2.2.

2.2.1 Control overhead due to the FIB inspection protocol

The number of IUs exchanged during the implementation of the FIB inspection protocol includes those carried by the handover initiation message (i.e, H_I), the set of messages exchanged between SDC and network access point (i.e. F_{req}, F_{res}, and F_{up}), and the set of messages exchanged with the ϵ_i neighbors of each $i-$th router belonging to $\mathcal{D}$, excepting $r_{d_{old}}$. Therefore, by also considering the details reported in Table 2.1, the average control overhead due to the FIB inspection protocol, $\bar{O}_{FI}$, is equal to:

$$\begin{aligned}
\bar{O}_{FI} &= \frac{1}{E[\Delta t]}\Bigg(E\left[H_I d_{c,sdc}\right] + E\left[(F_{req} + F_{res} + F_{up}) d_{c,sdc}\right] + \\
&\quad + E\left[\sum_{\substack{i=0 \\ i \ni' r_i \in \mathcal{D}}}^{d_{old}-1} \sum_{j=1}^{\epsilon_i} (F_{req} + F_{res} + F_{up}) d_{r_{ij},sdc}\right]\Bigg) = \\
&= \frac{2v}{\pi r}\bar{d}\left(4W + 1 + (3W+1)\bar{d}\bar{\epsilon}\right)\frac{IUs}{s}. \qquad (2.1)
\end{aligned}$$

Whereas, it can be proven that $E\left[\sum_{i=1}^{d} \epsilon_i | d\right] = E\left[\sum_{i=1}^{d} \epsilon_i\right] = \sum_{i=1}^{d} E[\epsilon_i] = d\bar{\epsilon}$. Moreover, given the probability distribution of the shortest path length, i.e., $p(d)$, it holds that $E\left[d\bar{\epsilon}\right] = \sum_d d\bar{\epsilon}p(d) = \bar{\epsilon}\bar{d}$.

2.2.2 Control overhead due to the neighbor inspection protocol

The number of IUs exchanged during the implementation of the neighbor inspection protocol includes those carried by the handover initiation message (i.e., H_I), the messages exchanged between the SDC and the network access point of the consumer (i.e., N_{req}, N_{res}, and N_{up}), and the messages exchanged with the $k_i - 1$ neighbors of each i-th router belonging to $\mathcal{D}$, excepting $r_{d_{old}}$. Therefore, by also considering the details reported in Table 2.1, the average control overhead due to the neighbor inspection protocol, $\bar{O}_{NI}$, is equal to:

$$\begin{aligned}
\bar{O}_{NI} &= \frac{1}{E[\Delta t]}\Bigg(E\left[H_I d_{c,sdc}\right] + E\left[(N_{req} + N_{res} + N_{up}) d_{c,sdc}\right] + \\
&\quad + E\left[\sum_{\substack{i=0 \\ i \ni' r_i \in \mathcal{D}}}^{d_{old}-1} \sum_{j=1}^{k_i-1} (N_{req} + N_{res} + N_{up}) d_{r_{ij},sdc}\right]\Bigg) = \\
&= \frac{2v}{\pi r}\bar{d}(4W + 3W\bar{d}(\bar{k}-1))\frac{IUs}{s}. \qquad (2.2)
\end{aligned}$$

2.2.3 Control overhead due to SDN-based re-sync protocol

With respect to the attachment process, the SDN-based re-sync protocol envisages the delivery of IUs through the handover completed message (i.e, H_C) and the set of messages managed by the SDC on the control plane to retrieve the last content name (i.e. $L_{SDN,req}$, $L_{SDN,res}$, and $L_{SDN,mes}$). Therefore, the average control overhead due to SDN-based re-sync protocol, $\bar{O}_{SDN}$, is equal to:

$$\begin{aligned} \bar{O}_{SDN} &= \frac{1}{E[\Delta t]} E\Big[H_C d_{c,sdc} + (L_{SDN,req} + L_{SDN,res}) d_{p,sdc} + L_{SDN,msg} d_{c,sdc}\Big] = \\ &= 4\bar{d}\frac{2v}{\pi r}\,\frac{IUs}{s}. \end{aligned} \tag{2.3}$$

2.2.4 Control overhead due to NDN-based re-sync protocol

Regarding the NDN-based re-sync protocol, the number of exchanged IUs includes contributions from the handover completed message (i.e., H_C) and the Interest and Data packets forwarded across the routers belonging to the new path (i.e., $L_{NDN,req}$ and $L_{NDN,res}$). Accordingly, the average control overhead due to the NDN-based re-sync protocol, $\bar{O}_{NDN}$, is equal to:

$$\begin{aligned} \bar{O}_{NDN} &= \frac{1}{E[\Delta t]} E\Big[H_C d_{c,sdc} + (L_{NDN,req} + L_{NDN,res}) d_{c,p}\Big] = \\ &= 3\bar{d}\frac{2v}{\pi r}\,\frac{IUs}{s}. \end{aligned} \tag{2.4}$$

2.3 Cross comparison in different network scenarios

To practically evaluate the control overhead in different scenarios, a network with a variable number of nodes, N ranges from 10 to 10000, is considered. The window size W is set to 1 and 10. The average number of faces stored within the FIB entry, $\bar{\epsilon}$, is set to $\bar{\epsilon} = \bar{k} - 1$. Moreover, in order to consider an urban scenario the average radius of the coverage area of network access points is set to $r = 150$ m and the average consumer speed is set to $v = 3$ km/h and $v = 50$ km/h. Figures 2.6 and 2.7 show the upper bound of average control overhead evaluated in different network conditions for each combination of protocols addressing detaching and attaching processes.

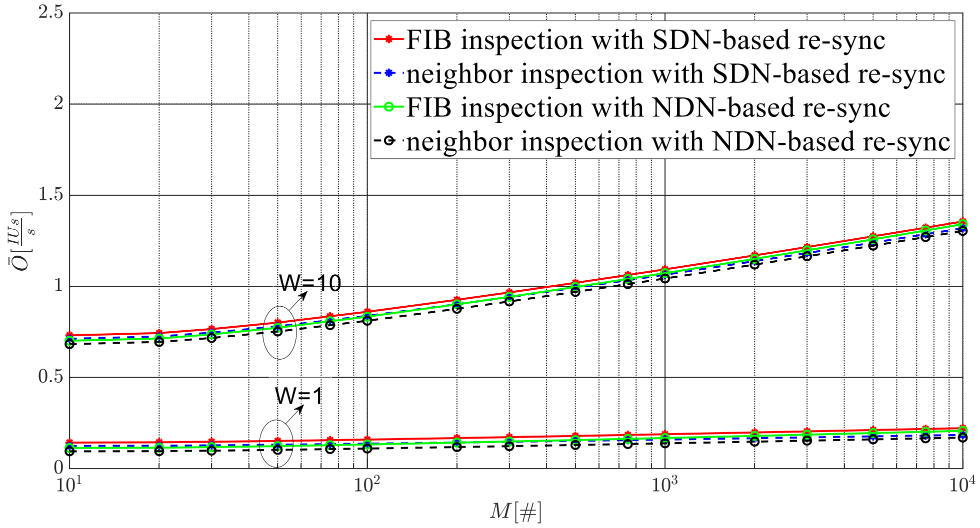


FIGURE 2.6: Control overhead evaluated when the average consumer speed is set to $v = 3$, varying the number of routers N in the topology.

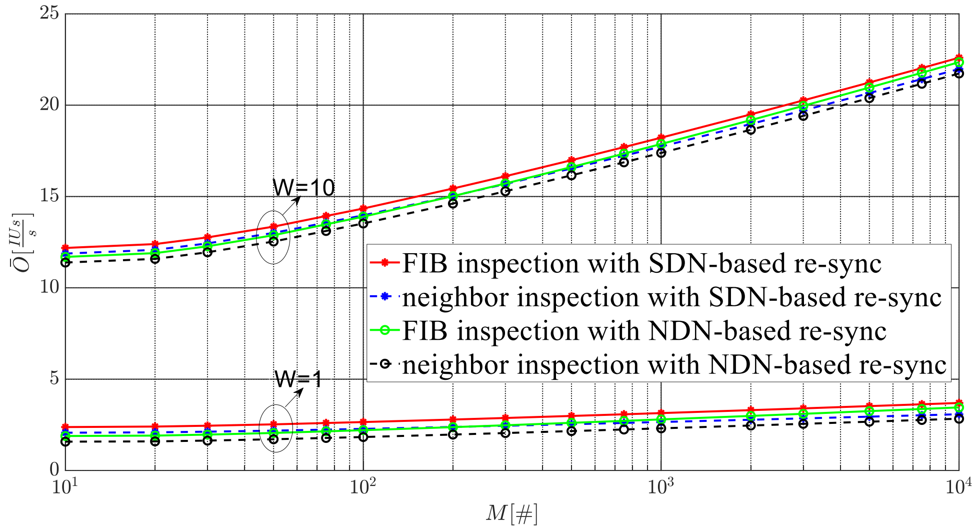


FIGURE 2.7: Control overhead evaluated when the average consumer speed is set to $v = 50$, varying the number of routers N in the topology.

Obtained results generally demonstrate that the total average control overhead increases with the number of routers in the network, the window size *W*, and the consumer speed. As well known, the average shortest path increases with the network size. Therefore, the higher the average shortest path, the higher the number of routers and neighbors to inspect. Regarding the sliding-window mechanism, the window size *W* influences the control overhead as well. In this case, the higher the window size, the higher the number of IUs exchanged during the implementation of protocols for the detachment procedure. The control overhead also grows with the average consumer speed v. This is because when a mobile consumer changes a higher number of network access points over time, it inevitably triggers many handover procedures. Accordingly, while the consumer speed increases, the amount of time protocols for both detachment and attachment processes increases too.

Regarding the combination of solutions used for both detachment and attachment mechanisms, it is evident how the adoption of neighbor inspection and NDN-based re-sync protocols achieves the best performance. This result is due to two main reasons. First, the neighbor inspection protocol requires the exchange of only PIT entries (FIB inspection protocol, instead, also involves FIB entries). Second, the NDN-based re-sync protocol envisages the interaction among only two network entities. On the contrary, the adoption of FIB inspection and SDN-based re-sync protocols always provides the highest control overhead.

Chapter 3

A Softwarized and MEC-Enabled Protocol Architecture Supporting Consumer Mobility in Information-Centric Networks

In order to solve the issues related to consumer mobility in Information-Centric Networks, this chapter extends and advances the proposal in Chapter 2 by illustrating a novel protocol architecture that successfully integrates and properly customizes the key functionalities of Information-Centric Networking, Multi-access Edge Computing, and Software Defined Networking paradigms. It envisages that (1) Multi-access Edge Computing assists mobile consumers in retrieving data, while transparently managing the information-centric communication primitives and recovering the synchronization with the remote producer after the handover, (2) Software-Defined Controllers dynamically configure forwarding functionalities, and (3) Information-Centric Networking enables efficient data dissemination and delivers network control instructions.

The impact of the devised protocol architecture on the communication overhead is analytically formulated and evaluated in scenarios with different topology, mobility, application settings, and number of mobile consumers.

3.1 The proposed protocol architecture

This work focuses on a multi-hop wireless mesh network (see Figure 3.1). Many network attachment points are randomly located within a given geographical area and provide wireless connectivity to mobile consumers. According to the multi-access paradigm, the network may support heterogeneous communication technologies. But, at the higher layers of the protocol stack, all the nodes interact with each other through NDN, which transparently operates with respect to the technical details characterizing the underlying communication technologies. This is also valid for the control messages, that are exchanged within the multi-hop wireless mesh network (i.e., between the SDC and the routers) via NDN communication primitives. Each network attachment point hosts a MEC entity, whose goal is to assist mobile consumers during the data retrieving process while ensuring a seamless and transparent interface with the NDN network. On the other hand, SDN is integrated within the whole protocol architecture for dynamically configuring NDN network functionalities. In summary, the conceived protocol architecture implements Subscription, Data Exchange, Attachment, Inspection, and Re-synchronization procedures. All of them are implemented at the higher layers of the protocol stack. Therefore, they do not depend on the kind of communication technologies available at the wireless interface (in fact, heterogeneous technologies are transparently managed by NDN).

In line with the issues described in Chapter 1, this section will use the following key terminologies: *unuseful data* and *lost data*. The term "unuseful data" refers to the contents delivered, after the handover, across stale disjoint links due to stale forwarding information. The delivery of such contents generates a waste of bandwidth within stale disjoint links, thus hindering network performance. On the other hand, the term "lost data" describes the set of contents lost by the mobile consumer during the handover procedure.

The list of entities involved in each procedure, as well as the different exchanged messages, are reported in Table 3.1.

3.1.1 The protocol architecture with a single SDN controller

For multi-hop wireless mesh networks made up of a limited number of network attachment points, it is conceivable to assume the presence of a single and centralized SDC. In this case, the conceived procedures are implemented as discussed below.

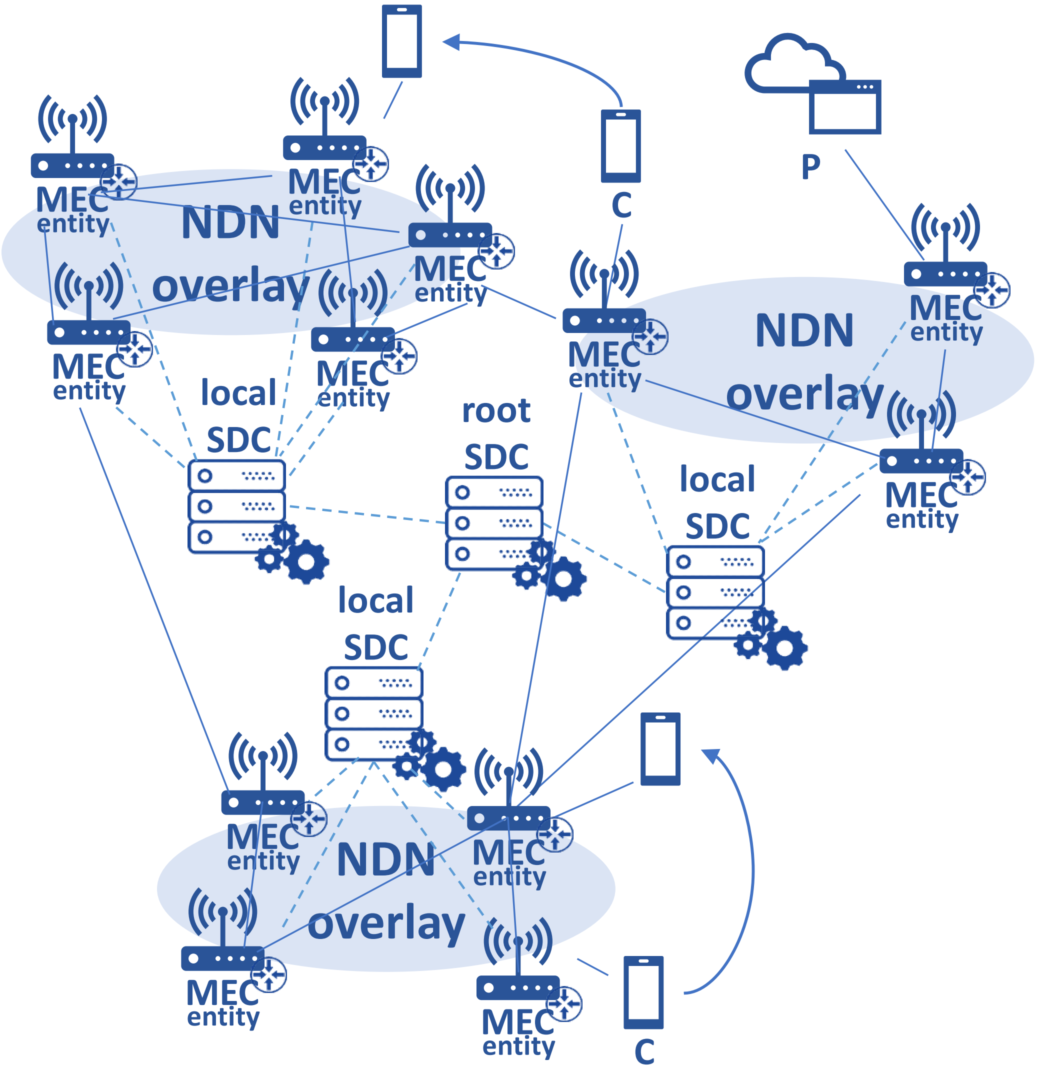


FIGURE 3.1: The reference network architecture.

Subscription procedure

The interaction between the mobile consumer and the MEC entity follows the publish-subscribe mechanism. Once the consumer attaches to a new network attachment point, it delivers to the corresponding MEC entity a Subscription Request, which indicates the topic-name of its interest. The request is encoded with an HTTP POST message and establishes a high-level connection (a web socket, for instance) between the mobile consumer and the MEC entity. Then, the MEC entity executes the Attachment procedure (as discussed in Section 3.1.1), the Re-synchronization procedure (as discussed in Section 3.1.1), and the Data Exchange procedure (as described in Section 3.1.1). Differently from what happens in the rest of the network, mobile consumer and MEC entity do not communicate with each other by using information-centric primitives. Indeed, no Interest and Data packets are issued at this

TABLE 3.1: An overview of procedures belonging to the conceived protocol architecture.

Procedure	Goal	Involved entities	Message name	Type of message	Average size
Subscription	Subscribe to a topic name	Consumer and MEC entity	Subscription Request	HTTP (POST)	S_R
Data Exchange	Retrieve desired data	MEC entities and Producer	Request	Interest packet	I_{INT}
			Response	Data packet	S_D
Attachment	Announce consumer attachment	SDC and MEC entities	Attachment Notification	Interest packet	I_{AN}
			Attachment Notification Confirmation	Data packet	D_{AN}
Inspection	Update PIT tables in network routers	SDC and MEC entities	Face Remove	Interest packet	I_{FR}
			Face Remove Confirmation	Data packet	D_{FR}
Re-synchronization	Announce the latest content generated	MEC entities and Producer	Re-sync Request	Interest packet	I_{RS}
			Re-sync Response	Data packet	D_{RS}

stage of the protocol. This makes the implementation of the end-user application independent from the protocol stack available beyond the network attachment point, i.e., within the multi-hop wireless mesh network.

Attachment procedure

The Attachment procedure is triggered by a MEC entity, upon the reception of a subscription request. Specifically, the MEC entity notifies the new location of the consumer and the topic name of its interest to the SDC. This is done by issuing the Attachment Notification message, encoded with an Interest packet having the content name set to $ndn : //Attachment - Notification/[MEC - entity - name]/[topic - name]/[consumer - name]$. The SDC stores this information within a local database and releases a Data packet of confirmation, namely Attachment Notification Confirmation. Note that in case of handover, the SDC stores in its local database information related to the previous and the new network attachment points. Figure 3.2 reports the message sequence chart describing the Attachment procedure.

Re-synchronization procedure

Before retrieving the contents requested by the mobile consumer, the new MEC entity must implement the Re-synchronization procedure, which represents a fundamental task for the following two reasons. First, the MEC entity does not know the ID of the latest content to request. Second, the interaction with the remote producer is useful to acquire the IDs of contents generated during the handover, especially after a long period of network detachment. As a result, the re-synchronization will ensure that the MEC entity will request up-to-date contents and, if necessary, retrieve lost data (i.e.,

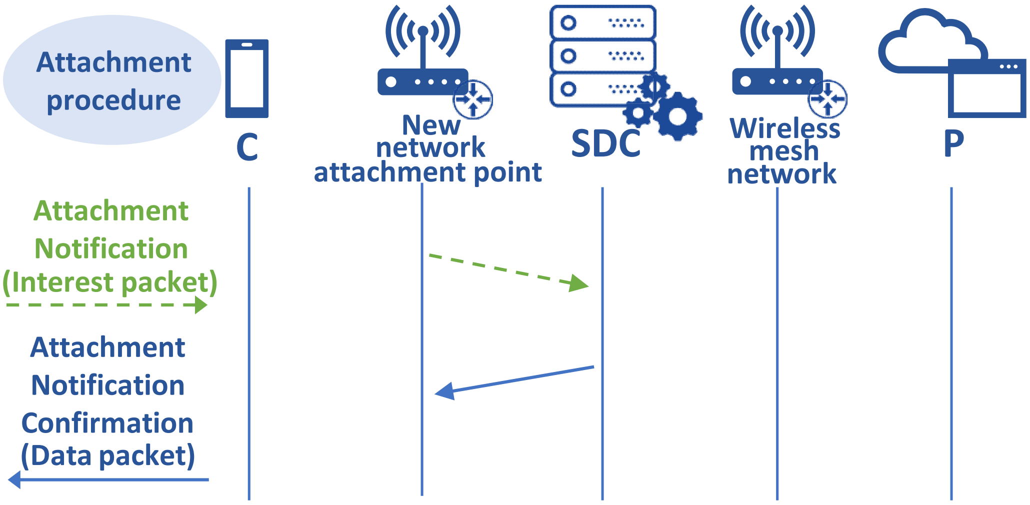


FIGURE 3.2: Messages exchanged during Attachment procedure.

contents which the mobile consumer was unable to receive during the handover). Indeed, the new MEC entity serving the consumer sends an Interest packet with the name set to $ndn : //[topic - name]/LAST/[timestamp]$. The producer answers with a Data packet containing the name of the latest content available for that topic-name. Figure 3.3 shows the message sequence chart describing the Re-synchronization procedure.

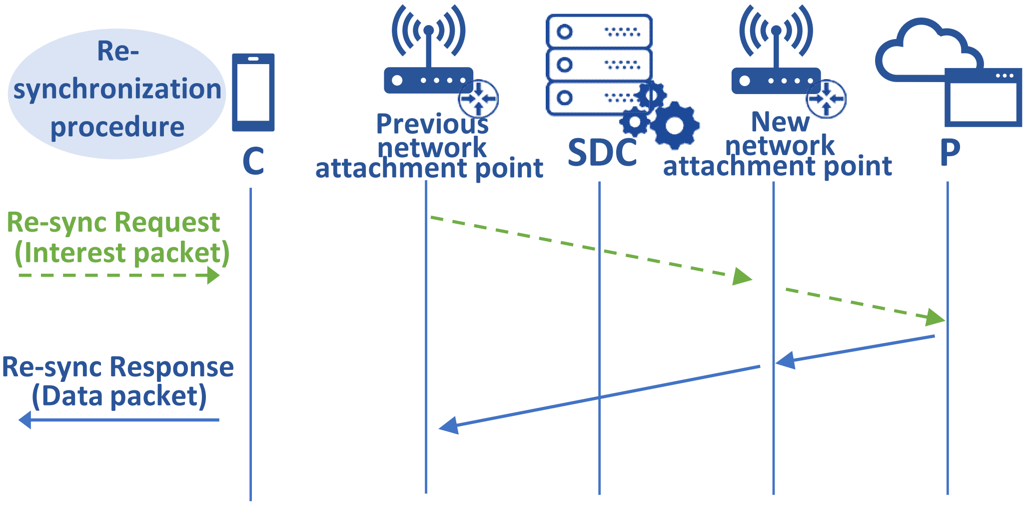


FIGURE 3.3: Messages exchanged during the Re-synchronization procedure.

Data Exchange procedure

In line with the NDN logic, the producer exposes real-time data under a specific topic-name. To simplify, a generic data can be identified with the content name $ndn : //[topic - name]/\#id$. During the Data Exchange procedure, the MEC entity starts retrieving new contents belonging to the subscribed topic by exploiting the conventional request-response communication scheme offered by NDN. Each request is implemented through an Interest packet, namely Request. Every time a new Response (implemented through a Data packet) is received, the MEC entity releases a new Request. Figure 3.4 reports the message sequence chart describing the Data Exchange procedure.

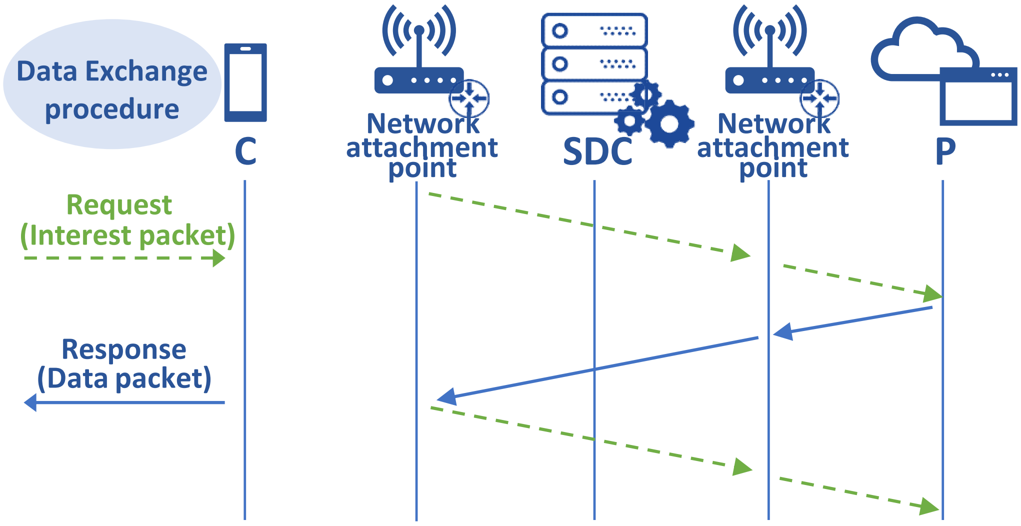


FIGURE 3.4: Messages exchanged during the Data Exchange procedure.

Inspection procedure

The Data Exchange procedure defines a new path between the mobile consumer and the producer. According to the NDN communication principles, the new request does not overwrite neither PIT nor FIB tables of the final part of the path. Instead, both communication paths (i.e., the one established between the new MEC entity and the remote producer after the handover and the other one established between the previous MEC entity and the same producer before the handover) remain active till the related PIT entries expire. To better manage mobility and real-time data delivery, the work presented in [12] suggests to set the lifetime of a PIT entry to the average amount of time between the generation of two consecutive contents belonging to the same topic name. Indeed, the stale disjoint path may remain active also after a long period of network detachment experienced by the mobile consumer in the case of non-overlapping cells. As a result, contents requested before the handover could be forwarded to the old location because of the wrong information stored in stale disjoint links. This inevitably brings to an unexpected waste of bandwidth. The SDC is now responsible for updating forwarding information stored within routers belonging to the stale disjoint path, i.e., by deleting the wrong PIT entries. This task is implemented through an iterative approach, namely Inspection procedure. Based on the way the path between the MEC entity serving the consumer and the remote producer is learned, two possible implementations are designed herein: Neighbor Inspection and Router Inspection. In both cases, the SDC starts investigating the previous network attachment point, whose details are stored within the local database (as anticipated in Section 3.1.1). Any control message is delivered across the network through information-centric primitives and processed following POF instructions. Figure 3.5 shows the message sequence chart describing the generic Inspection procedure.

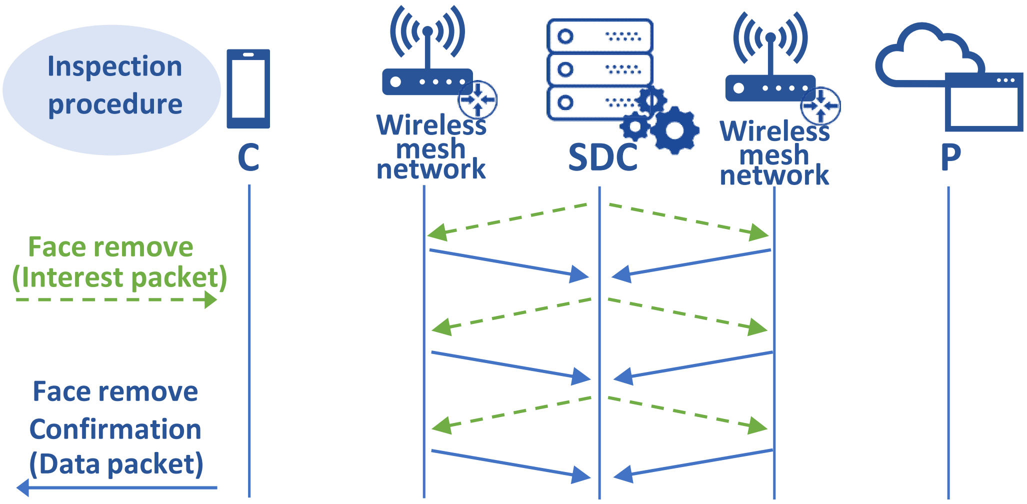


FIGURE 3.5: Messages exchanged during the Inspection procedure.

Neighbor Inspection) This procedure assumes that the SDC does not know a priori the path established between the MEC entity serving the consumer and the remote producer. In fact, even if the SDC knows the overall network topology (i.e., how and by which interface NDN routers are connected), it does not have any idea about the path followed by a specific Interest packet to reach the remote producer. As a consequence, it will learn that path by inspecting the network routers (first) and it will remove stale forwarding information from the routers of the discovered path (then). The inspection starts from the router corresponding to the previous consumer location (i.e., the node where the consumer was attached before the handover) and potentially ends when the router which the producer is attached to is reached. At every round, the SDC sends a Face Remove message to the investigated router. This message represents an Interest packet with the content name set to $ndn : //Face - Remove/[node - name]/[topic - name]/[face]$. The content name reports the topic-name of interest for the consumer and the face that should have received the request (which is set by the SDC at every step of the procedure, according to the known network topology). The router that receives the Face Remove message compares the topic-name of pending requests with those stored in the entries of its PIT table. PIT entries that match, at the same time, the topic-name of pending requests and the face indicated by the SDC are firstly selected. Then, those containing only the face indicated by the SDC are deleted. The presence of more than one face in the other PIT entries demonstrates that the considered contents have been requested by other consumers from other paths. In this case, the router just erases the face indicated by the SDC from these entries. In the end, the router sends back to the SDC a Data packet, namely Face Remove Confirmation, notifying the list of content names effectively deleted from the PIT table. This way, the SDC learns which router had stale information. Then, the SDC contacts all the neighbors of the router which answered with a non-empty packet at the previous round. The SDC is able to accomplish this task because it knows the overall network topology. Indeed, the SDC sends the Face Remove message to all the identified neighbors and indicates, through the content name stored within the Interest packet, the face connecting the investigated router with the previous one and the topic-name of pending requests. The neighbors implement the same tasks discussed before and notify to the SDC of the list of deleted entries. The SDC iterates the protocol until it receives empty Face Remove Confirmation messages only, which implies that the SDC removed all the stale information across the stale disjoint path.

A concrete example is depicted in Figure 3.6, showing a mobile consumer moving from R_1 to R_6.

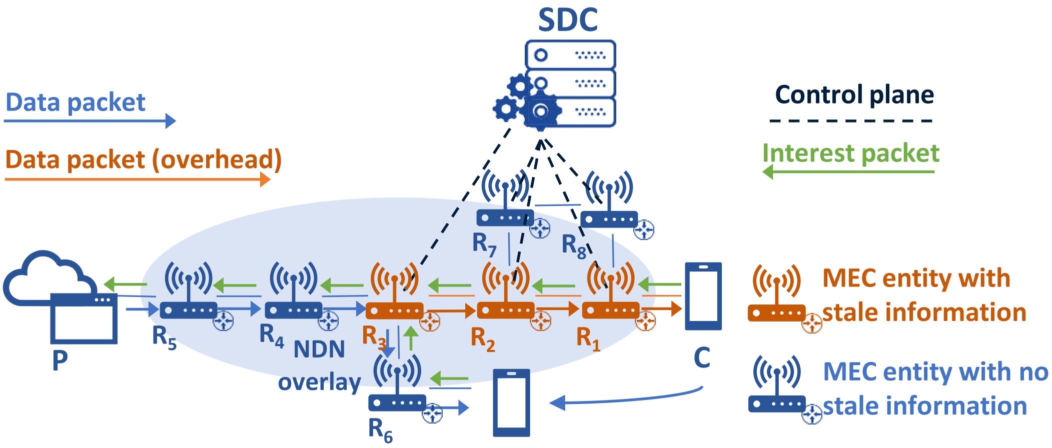


FIGURE 3.6: An example showing the Inspection procedure.

The Inspection procedure starts after the Re-synchronization procedure, when the mobile consumer is attached to R_6. The local SDC sends a Face Remove message to R_1, which sent the Attachment Notification message. The Face Remove message stores the name of the requested content and the interface which connects R_1 to the mobile consumer. R_1 compares the information received within the Face Remove message with its PIT table and answers to the local SDC with a Face Remove Confirmation message, storing the content name of the requested content and the interface related to the deleted PIT entry. Then, the local SDC sends a Face Remove message to the 1-hop neighbors of R_1. The message stores the content name related to the deleted PIT entry and the face which connects the inspected neighbor to R_1. Among all the inspected neighbors, only R_2 has PIT entries matching both the content name and the face, acknowledging that the content request came from R_1. Then, R_2 answers with a Face Remove Confirmation message to the local SDC. Finally, the SDC sends a Face Remove message to the 1-hop neighbors of R_2. Again, the Face Remove message carries the content name of the requested content and the face which connects the inspected neighbor to R_2. This time, R_3 finds PIT entries that also include other faces (i.e., the face that connects R_3 to R_6) and answers to the SDC with an empty Face Remove Confirmation message. The local SDC stops the implementation of the Inspection procedure, as it received only empty Face Remove Confirmation messages. Definitively, all stale information has been removed.

Router Inspection) The Router Inspection approach assumes that the SDC knows the path established between the MEC entity serving the consumer and the remote producer. Face Remove messages are not sent to all the neighbors anymore. Instead, the SDC iteratively inspects routers of the stale disjoint path, which may contain stale information. Similarly to the Neighbor Inspection approach, it sends a Face Remove message (i.e., an Interest packet whose content name is set to $ndn : //Face - Remove/[node - name]/[topic - name]/[face]$) to the inspected router, asking it to compare the information stored in the content names with its PIT entries. Then, the router sends back a Face Remove Confirmation message announcing content names for deleted entries. Accordingly, the SDC stops when it receives an empty Face Remove Confirmation message, meaning that it removed all stale information from the stale disjoint path.

Figure 3.6 can still be used to formulate a concrete example. When the mobile consumer attaches to R_6, after the Re-synchronization procedure, the local SDC sends a Face Remove message to R_1. The Face Remove message stores the content name of the requested content and the interface which connected R_1 to the mobile consumer. R_1 matches the received information with its PIT table and answer to the local SDC with a Face Remove Confirmation message, storing the content name of the requested content and the interface related to the deleted PIT entry. Then, the local SDC inspects the next router belonging to the path established between the MEC entity serving the consumer and the producer (i.e., R_2). The message stores the content name related to the deleted PIT entry and the face which connects R_2 to R_1. Then, R_2 answers with a Face Remove Confirmation message to the local SDC. Finally, the SDC sends a Face Remove message to R_3, that answers with an empty Face Remove Confirmation message. The local SDC stops the implementation of the Inspection procedure, as all stale information have been removed.

3.1.2 The protocol architecture with multiple SDN controllers

As expected, scalability grows in importance as the workload on the SDC increases. Thus, the execution of several complex operations on densely populated networks requires sophisticated solutions that inevitably counteract performance degradation. In large-scale deployments, the proposed protocol architecture could leverage multiple SDN controllers, organized according to a hierarchical model [33]: a root controller holds network-wide information and coordinates local SDN controllers; each local SDC manages a specific set of network attachment points belonging to its domain and holds information on attached consumers. In the case the stale disjoint path passes through more domains, the conceived protocol architecture must be modified in both the Attachment and Inspection procedures.

Revised Attachment procedure in case of multiple SDN controllers

The Attachment procedure starts when the mobile consumer attaches to a new network attachment point. The reference MEC entity notifies the local SDC with an Attachment Notification message. The local SDC apprehends that the consumer moved from another domain because it does not contain any information in its local database. As a consequence, the new local SDC polls the root controller by announcing the handover event. In turn, the root controller polls the local controllers notifying the name of the mobile consumer. Then, the local SDC which previously held information on the consumer answers to the root controller and starts the Inspection procedure.

Revised Inspection procedure in case of multiple SDN controllers

The local SDC that previously held information on the consumer starts the inspection from the router hosting the MEC entity where the consumer was attached before the handover. The procedure follows the same scheme detailed in Section 3.1.1 until a router belonging to a different domain is reached. In this case, the controller that started the Inspection procedure sends the name of the last router inspected to the root controller. The latter notifies the reference local controller about the name of the router it should start inspecting from. Then, the procedure continues as detailed in Section 3.1.1 until all stale information is removed.

3.2 Modeling the communication overhead

According to the scientific contribution published in [12], consumer mobility in an NDN network inevitably brings to a communication overhead due to the exchange of control messages and the dissemination of Data packets across the links of the stale disjoint path. Even though SDN is introduced for reducing the waste of bandwidth due to the dissemination of Data packets across these links, new control messages need to be exchanged among MEC

TABLE 3.2: List of model symbols.

Symbol	Description
Δt_c	Cell residence time of the consumer
L	Number of mobile consumers interested in the same contents
v	Average consumer speed
r	Average cell coverage radius
τ	Round Trip Time of the Network
δ	Processing time
T_D	Interarrival time
k	Average number of neighbors per router
A	Average number of routers on the stale disjoint path
d	Average shortest path between two nodes in the network
D	Average number of active links in the network
S_R	Packet size for Subscription Request
I_{INT}	Packet size for Request
S_D	Packet size for Response
I_{AN}	Packet size for Attachment Notification
D_{AN}	Packet size for Attachment Notification Confirmation
I_{FR}	Packet size for Face Remove
D_{FR}	Packet size for Face Remove Confirmation
I_{RS}	Packet size for Re-sync Request
D_{RS}	Packet size for Re-sync Response

entities, routers, and the SDCs. Indeed, the analysis of the communication overhead initially provided in [12] must be revised. The goal of this Section is to formulate analytical models able to quantify the communication overhead produced by the conceived protocol architecture in the presence of one or more mobile consumers. A summary of the main symbols adopted below is presented in Table 3.2.

In line with [12], the communication overhead is formally defined as the total amount of bytes related to messages on both the control plane and the data plane, exchanged at the application layer on the links of the multi-hop wireless mesh network in a unit of time. It is independent of the communication technology implemented at the lower layer of the protocol stack, which may induce packet fragmentation at the link level. As expected, Subscription, Attachment, Re-synchronization, Data Exchange, and Inspection procedures provide different contributions to the communication overhead due to control messages. Unuseful data, instead, still refers to the amount of Data packets delivered through the links of the stale disjoint path before that the Inspection procedure deleted all the wrong forwarding information in PIT entries. Indeed, it is important to remark that the conducted analysis only considers the signaling, and the corresponding overhead, related to the management of consumer mobility, as any other control communication remains unchanged and does not influence the Key Performance Indicators (KPIs) investigated in Section 3.3. Also, the analysis considers the scenario with a single SDC, but it can be simply extended for deployments having multiple controllers. Let $E[\Delta t_c]$ be the average cell residence time, that is the amount of time the consumer is connected to a specific MEC entity before attaching to a new one because of its mobility. Given the average cell radius, r, and the average consumer speed, v, the average cell residence time is set to $E[\Delta t_c] = \frac{\pi r}{2v}$ as in [89]. Moreover, d represents the average shortest path length between any node pair. At the application layer, the producer generates data packets with an average size equal to S_D according to the Poisson law with parameter $\frac{1}{T_D}$. Therefore, the average time interval between the generation of two consecutive content is equal to T_D.

The average cell residence time is used as the reference observation time for calculating the communication overhead. Moreover, let O_{SUB}, O_{INT}, O_{AT}, O_{RS}, O_{IN}, and O_{DATA} be the contributions to the communication overhead due to the subscription request sent by the mobile consumer to the reference MEC instance, the set of Interest packets issued by the MEC entity to retrieve contents during the Data Exchange procedure, control messages exchanged during the Attachment procedure, messages to restore the synchronization exchanged during the Re-synchronization procedure, control messages exchanged during the Inspection procedure, and unuseful Data, respectively. Definitively, the average communication overhead can be formally defined as:

$$\bar{O} = \frac{E[O_{SUB} + O_{INT} + O_{AT} + O_{RS} + O_{IN} + O_{DATA}]}{E[\Delta t_c]}, \tag{3.1}$$

where $E[\cdot]$ refers to the expectation operator.

3.2.1 Communication overhead expected in scenarios with a single consumer

With reference to a scenario with a single consumer, Theorem 1 and Theorem 2 formulate the communication overhead due to the proposed protocol architecture when the Neighbor Inspection and Router Inspection algorithms are implemented, respectively.

Theorem 1. *Let d, Δt_c, T_D, k, A, and τ be the average shortest path length, the average cell residence time of the mobile consumer, the time interval between the production of two consecutive contents, the average number of 1-hop neighbors per router, the average number of routers on the stale disjoint path, and the round trip time of the network. Considering that the generation of packets follows a Poisson law with parameter $\frac{1}{T_D}$, the average communication overhead registered when the Neighbor Inspection algorithm is implemented is equal to:*

$$\begin{aligned}
\bar{O}\Big|_{N.\ Insp} &= \frac{S_R}{\Delta t_c} + \\
&+ \frac{d I_{INT}}{\Delta t_c}\left(1 + \frac{\Delta t_c}{T_D}\right) + \\
&+ \frac{d}{\Delta t_c}(I_{AN} + D_{AN}) + \\
&+ \frac{d}{\Delta t_c}(I_{RS} + D_{RS}) + \\
&+ \frac{d}{\Delta t_c}(I_{FR} + D_{FR})((A-1)(k-1)+2) + \\
&+ \frac{S_D}{\Delta t_c}\left(A + e^{-\frac{\tau}{T_D}} - \frac{1 - e^{-(A+1)\frac{\tau}{T_D}}}{1 - e^{-\frac{\tau}{T_D}}}\right),
\end{aligned} \tag{3.2}$$

where S_R, I_{INT}, S_D, I_{AN}, D_{AN}, I_{FR}, D_{FR}, I_{RS}, and D_{RS} are the average size of Subscription Request, Request, Response, Attachment Notification, Attachment Notification Confirmation, Face Remove, Face Remove Confirmation, Re-sync Request, and Re-sync Response messages, respectively.

Proof. The consumer sends a Subscription Request message to the MEC entity hosted by its reference network attachment point. The related contribution to the communication overhead is simply equal to:

$$E[O_{SUB}] = E\left[S_R\right] = S_R. \tag{3.3}$$

Then, during the Data Exchange procedure, the related MEC entity starts pulling the contents to retrieve by sending an Interest packet. The consumer releases a new Interest packet every time a new Data packet is received. This happens every T_D, till the end of the average cell residence time $E[\Delta t_c] = \frac{\pi r}{2v}$. Let I_{INT} and $d_{M \rightarrow P}$ be the size of the Interest packet and the distance between the MEC entity implementing the Data Exchange procedure and the router which the producer is attached to, respectively. Then, as already demonstrated in [12], the average communication overhead due to the data exchange, $E[O_{INT}]$, is equal to:

$$\begin{aligned} E[O_{INT}] &= E\left[d_{M\rightarrow P} I_{INT}\left(1+\frac{\Delta t_c}{T_D}\right)\right] = E\left[d_{M\rightarrow P}\right] I_{INT}\left(1+\frac{\Delta t_c}{T_D}\right) = \\ &= d I_{INT}\left(1+\frac{\Delta t_c}{T_D}\right). \end{aligned} \tag{3.4}$$

Note that T_D and I_{INT} have been introduced as average values. Therefore they are considered as constant terms by the expectation operator $E[\cdot]$. Differently, $E[d_{M\rightarrow P}] = d$.

During the Attachment procedure, two control messages are exchanged: the Attachment Notification message (with a size equal to I_{AN}) is sent by the MEC entity that started the Attachment procedure to the remote SDC; the Attachment Notification Confirmation message (with a size equal to D_{AN}) is sent by the SDC to acknowledge the new position of the consumer. Note that $d_{M'\rightarrow C}$ represent the distance between the MEC entity serving the consumer after the handover and the SDC. Therefore, by assuming $E[d_{M'\rightarrow C}] = d$, the average communication overhead due to the Attachment procedure is equal to:

$$\begin{aligned} E[O_{AT}] &= E[d_{M'\rightarrow C}(I_{AN}+D_{AN})] = E[d_{M'\rightarrow C}](I_{AN}+D_{AN}) = \\ &= d(I_{AN}+D_{AN}). \end{aligned} \tag{3.5}$$

The execution of Re-synchronization procedure envisages the exchange of Re-sync Request and Re-sync Response messages. Its contribution to the communication overhead is $O_{RS} = (I_{RS}+D_{RS})d_{M'\rightarrow P}$. Considering average shortest path length $E[d_{M'\rightarrow P}] = d$ and the size of the aforementioned messages (I_{RS} and D_{RS}), the average communication overhead due to the Re-synchronization procedure is equal to:

$$\begin{aligned} E[O_{RS}] &= E\left[d_{M'\rightarrow P}(I_{RS}+D_{RS})\right] = E\left[d_{M'\rightarrow P}\right](I_{RS}+D_{RS}) = \\ &= d(I_{RS}+D_{RS}). \end{aligned} \tag{3.6}$$

A more complex discussion should be done for the Neighbor Inspection procedure. Let r_i be the $i-th$ router belonging to the stale disjoint path.

r_1, for instance, represents the router hosting the MEC entity where the consumer was previously attached. Let n_{ij} be the $j-th$ neighbor of r_i. To simplify the notation, it is assumed that the average number of routers belonging to the stale disjoint path is equal to A. Moreover, I_{FR} and D_{FR} represent the size of Face Remove and Face Remove Confirmation messages, respectively. The distance between r_i and the SDC is denoted with $d_{r_i \to C}$. Similarly, the distance between the node n_{ij} and the SDC is denoted with $d_{n_{ij} \to C}$. The Inspection procedure starts investigating the PIT tables of the first node of the path and its neighbors. The resulting communication overhead is equal to $d_{M \to C}(I_{FR} + D_{FR}) + \sum_{j=1}^{k} d_{n_{1j} \to C}(I_{FR} + D_{FR})$. Starting from r_2, only k-1 neighbors of the considered router of the disjoint path are investigated. In fact, the current node and the previous router of the path (i.e., a neighbor of the current node) have been already investigated one step before. Thus, each new interaction of the protocol produces a contribution to the communication overhead equal to $\sum_{j=1}^{k-1} d_{n_{ij} \to C}(I_{FR} + D_{FR})$. Without loss of generality, it is assumed that the Inspection procedure investigates all the routers of the stale disjoint path. The SDC must inspect every single router on the stale disjoint path to remove stale information and avoid removing up-to-date forwarding information from aggregated PIT entries. In fact, even if the SDC knows the router where the old and the new paths join, removing the face related to the old requests of the mobile consumer could hinder the requests of other consumers that are aggregated on the other routers of the stale disjoint path. In this case, the contribution to the communication overhead due to the Inspection procedure is equal to:

$$\begin{aligned}
E[O_{IN}]\Big|_{\text{N. Insp}} &= E\left[d_{M\rightarrow C}(I_{FR}+D_{FR})+\sum_{j=1}^{k} d_{n_{1j}\rightarrow C}(I_{FR}+D_{FR})+\right.\\
&\left.+\sum_{i=2}^{A-1}\sum_{j=1}^{k-1} d_{n_{ij}\rightarrow C}(I_{FR}+D_{FR})\right]=\\
&= (I_{FR}+D_{FR})E\left[d_{M\rightarrow C}\right]+(I_{FR}+D_{FR})E\left[\sum_{j=1}^{k} d_{n_{1j}\rightarrow C}\right]+\\
&+(I_{FR}+D_{FR})E\left[\sum_{i=2}^{A-1}\sum_{j=1}^{k-1} d_{n_{ij}\rightarrow C}\right]=\\
&= (I_{FR}+D_{FR})d+(I_{FR}+D_{FR})kd+\\
&+(I_{FR}+D_{FR})(A-2)(k-1)d=\\
&= d(I_{FR}+D_{FR})(1+k+(A-2)(k-1))=\\
&= d(I_{FR}+D_{FR})((A-1)(k-1)+2). \qquad (3.7)
\end{aligned}$$

Although the goal of the SDC is to delete stale information of PIT entries, Data packets may still be forwarded across the links belonging to the stale disjoint path before the end of the Inspection procedure. Let τ and δ be the round trip time and the time needed to process the Face Remove message. Since $\delta << \tau$, it is assumed that δ is negligible with respect to τ. The stale disjoint path is made up by A routers and the stale forwarding information of the $i-th$ router of the stale disjoint path can be deleted after $i\tau$ from the beginning of the cell residence time. Therefore, the contribution to the communication overhead due to the delivery of unuseful data is equal to $S_D(A-1)$ if a new content is generated within the time interval 2τ from the begin of the cell residence time (with probability $P(0 \leq t < 2\tau)$), $S_D(A-2)$ if a new content is generated in the interval between 2τ and 3τ (with probability $P(2\tau \leq t < 3\tau)$), from the beginning of the cell residence time, and so on. No unuseful Data packets are exchanged if the new content is generated after $A\tau$ from the beginning of the cell residence time. Considering that the contents are generated according to the Poisson law with parameter $\frac{1}{T_D}$, it holds that $P(0 \leq t < 2\tau) = P(0 \leq t < \tau) + P(\tau \leq t < 2\tau) = 1 - e^{-\frac{2\tau}{T_D}}$ and $P(i\tau \leq t < (i+1)\tau) = (1 - e^{-\frac{(i+1)\tau}{T_D}}) - (1 - e^{-\frac{i\tau}{T_D}})$. Therefore, $E[O_{DATA}]$ can be expressed as:

$$\begin{aligned}
E[O_{DATA}] &= S_D\Bigg(P(0 \leq t < \tau)(A-1) + \\
&+ \sum_{i=1}^{A-1} P(i\tau \leq t < (i+1)\tau)(A-i)\Bigg) = \\
&= S_D\Bigg((1-e^{-\frac{\tau}{T_D}})(A-1) + \sum_{i=1}^{A-1}((1-e^{-(i+1)\frac{\tau}{T_D}}) + \\
&-(1-e^{-i\frac{\tau}{T_D}}))(A-i)\Bigg) = \\
&= S_D\Bigg((1-e^{-\frac{\tau}{T_D}})(A-1) + \sum_{i=1}^{A-1} e^{-i\frac{\tau}{T_D}}(1-e^{-\frac{\tau}{T_D}})(A-i)\Bigg) = \\
&= S_D\Bigg((1-e^{-\frac{\tau}{T_D}})(A-1) + (1-e^{-\frac{\tau}{T_D}})A\sum_{i=1}^{A-1} e^{-i\frac{\tau}{T_D}} + \\
&-(1-e^{-\frac{\tau}{T_D}})\sum_{i=1}^{A-1} ie^{-i\frac{\tau}{T_D}}\Bigg) = \\
&= S_D\Bigg(A + e^{-\frac{\tau}{T_D}} - \frac{1-e^{-(A+1)\frac{\tau}{T_D}}}{1-e^{-\frac{\tau}{T_D}}}\Bigg). \qquad (3.8)
\end{aligned}$$

Now, by substituting (3.3), (3.4), (3.5), (3.6), (3.7), and (3.8) in (3.1), it is possible to prove the theorem.

□

Theorem 2. *Let d, Δt_c, T_D, A, and τ be the average shortest path length, the average cell residence time of the mobile consumer, the time interval between the production of two consecutive contents, the average number of routers on the stale disjoint path, and the round trip time of the network. Considering that the generation of packets follows a Poisson law with parameter $\frac{1}{T_D}$, the average communication overhead registered when the Router Inspection algorithm is implemented is equal to:*

$$\begin{aligned}
\bar{O}\Big|_{R.\ Insp} &= \frac{S_R}{\Delta t_c} + \\
&+ \frac{dI_{INT}}{\Delta t_c}\left(1 + \frac{\Delta t_c}{T_D}\right) + \\
&+ \frac{d}{\Delta t_c}(I_{AN} + D_{AN}) + \\
&+ \frac{d}{\Delta t_c}(I_{RS} + D_{RS}) + \\
&+ \frac{d}{\Delta t_c}(I_{FR} + D_{FR})(A-1) + \\
&+ \frac{S_D}{\Delta t_c}\left(A + e^{-\frac{\tau}{T_D}} - \frac{1 - e^{-(A+1)\frac{\tau}{T_D}}}{1 - e^{-\frac{\tau}{T_D}}}\right),
\end{aligned} \tag{3.9}$$

where I_{INT}, S_D, I_{AN}, D_{AN}, I_{FR}, D_{FR}, I_{RS}, and D_{RS} are the average size of Request, Response, Attachment Notification, Attachment Notification Confirmation, Face Remove, Face Remove Confirmation, Re-sync Request, and Re-sync Response messages, respectively.

Proof. The contribution to the communication overhead due to the Data Exchange, Attachment, and Re-synchronization procedures remains the same also in this case. Differently, the overhead due to the Inspection procedure must be revised. When the Router Inspection algorithm is implemented, only the routers of the stale disjoint path are investigated. The contribution to the communication overhead is equal to $d_{M\to C}(I_{FR} + D_{FR})$ (for the first node) and $d_{r_i \to C}(I_{FR} + D_{FR})$ (for the other ones). Therefore:

$$\begin{aligned}
E[O_{IN}]\Big|_{\text{R. Insp}} &= E\left[d_{M\to C}(I_{FR} + D_{FR}) + \sum_{j=2}^{A-1} d_{r_j \to C}(I_{FR} + D_{FR})\right] = \\
&= (I_{FR} + D_{FR})d + (I_{FR} + D_{FR})E\left[\sum_{j=2}^{A-1} d_{r_j \to C}\right] = \\
&= (I_{FR} + D_{FR})d + (I_{FR} + D_{FR})(A-2)d = \\
&= d(I_{FR} + D_{FR})(A-1).
\end{aligned} \tag{3.10}$$

Now, by substituting (3.3), (3.4), (3.5), (3.6), (3.8), and (3.10) in (3.1), the theorem is proved. □

3.2.2 Communication overhead expected in scenarios with multiple consumers

With reference to a scenario with multiple consumers, Theorem 3 and Theorem 4 formulate the communication overhead due to the proposed protocol when the Neighbor Inspection and the Router Inspection algorithms are implemented, in the multiple consumer scenario, respectively.

Theorem 3. *Let Δt_c, T_D, d, D, A, and L be the average cell residence time, the time interval between the production of two consecutive contents, the average shortest path length between any two nodes, the average number of active links in the network during the cell residence time, the average number of stale disjoint links, and the number of consumers requesting the same contents. Considering that the generation of packets follows a Poisson law with parameter $\frac{1}{T_D}$, the average communication overhead registered when the Neighbor Inspection algorithm is implemented is equal to:*

$$\begin{aligned}\bar{O}\Big|_{N.\ Insp} &= \frac{L}{\Delta t_c} S_R + \\ &+ \frac{L}{\Delta t_c}\left(dI_{INT} + \frac{\Delta t_c}{LT_D} DI_{INT}\right) + \\ &+ \frac{dL}{\Delta t_c}(I_{AN} + D_{AN}) + \\ &+ \frac{dL}{\Delta t_c}(I_{RS} + D_{RS}) + \\ &+ \frac{dL}{\Delta t_c}(I_{FR} + D_{FR})((A-1)(k-1)+2) + \\ &+ \frac{L}{\Delta t_c} S_D \left(A + e^{-\frac{\tau}{T_D}} - \frac{1 - e^{-(A+1)\frac{\tau}{T_D}}}{1 - e^{-\frac{\tau}{T_D}}}\right), \qquad (3.11)\end{aligned}$$

where I_{INT}, S_D, I_{AN}, D_{AN}, I_{FR}, D_{FR}, I_{RS}, and D_{RS} are the average size of Request, Response, Attachment Notification, Attachment Notification Confirmation, Face Remove, Face Remove Confirmation, Re-sync Request, and Re-sync Response messages, respectively.

Proof. The contribution to the communication overhead due to the Subscription, Attachment, Re-synchronization, and Inspection procedures remains the same as for the single consumer scenario. Differently, the overhead due to the Data exchange procedure has to be revised, starting from the re-definition of the average cell residence time. Given that L consumers are moving independently in an area covered by network attachment points, the cell residence time of the $i-th$ consumer can be modeled as an exponential random variable with parameter λ_i [89]. For the sake of simplicity, it is also assumed that a single user can change network attachment point at a time and all mobile consumers have the same parameter $\lambda_i = \lambda$. Let $P(\Delta t_{c,i} \geq T)$ be the probability that the residence time of the $i-th$ user $\Delta t_{c,i}$ is greater or equal than T, describing the probability that the $i-th$ consumer does not change its network attachment point during the time interval T. Then, the probability that L users do not change network attachment point during the time interval T, $P_L(\Delta t_L \geq T)$, is equal to:

$$P_L(\Delta t_L \geq T) = \prod_{i=1}^{L} P(\Delta t_{c,i} \geq T) = \prod_{i=1}^{L} e^{-\lambda T} = e^{-L\lambda T}. \tag{3.12}$$

The average cell residence time of the consumers in the network, i.e., the time between two consecutive handovers registered within the whole network, is equal to:

$$E[\Delta t_L] = \frac{1}{L\lambda} = \frac{\Delta t_c}{L}, \tag{3.13}$$

where Δt_c is the average cell residence time of a single consumer.

During the Data Exchange procedure, the related MEC entity starts pulling the contents to retrieve by sending an Interest packet, which aggregates with other requests from users interested in the same content. The consumer releases a new Interest packet every time a new Data packet is received. This happens every T_D, till the end of the average cell residence time $E[\Delta t_c] = \frac{\pi r}{2vL}$. Let I_{INT} and $d_{M\rightarrow P}$ be the size of the Interest packet and the distance between the MEC entity implementing the Data Exchange procedure and the router which the producer is attached to, respectively. Also, let D be the number of active links during the cell residence time (i.e, the links over which Interest and Data packets of all consumers interested in the same contents are exchanged on). Indeed, the average communication overhead due to the data exchange, $E[O_{INT}]$, is equal to:

$$E[O_{INT}] = dI_{INT} + \frac{\Delta t_c}{LT_D} DI_{INT} \tag{3.14}$$

By substituting (3.3), (3.5), (3.6), (3.7), (3.8), (3.13), and (3.14) in (3.1), the theorem is proved.

□

Theorem 4. *Let Δt_c, T_D, d, D, A, and L be the average cell residence time, the time interval between the production of two consecutive contents, the average shortest path length between any two nodes, the average number of active links in the network during the cell residence time, the average number of stale disjoint links, and the number of consumers requesting the same contents. Considering that the generation of packets follows a Poisson law with parameter $\frac{1}{T_D}$, the average communication overhead registered when the Router Inspection algorithm is implemented is equal to:*

$$\begin{aligned}
\bar{O}\Big|_{R.\ Insp} &= \frac{S_R}{\Delta t_c} + \\
&+ \frac{L}{\Delta t_c}\left(dI_{INT} + \frac{\Delta t_c}{LT_D} DI_{INT}\right) + \\
&+ \frac{dL}{\Delta t_c}(I_{AN} + D_{AN}) + \\
&+ \frac{dL}{\Delta t_c}(I_{RS} + D_{RS}) + \qquad (3.15) \\
&+ \frac{dL}{\Delta t_c}(I_{FR} + D_{FR})(A-1) + \\
&+ \frac{L}{\Delta t_c} S_D \left(A + e^{-\frac{\tau}{T_D}} - \frac{1 - e^{-(A+1)\frac{\tau}{T_D}}}{1 - e^{-\frac{\tau}{T_D}}}\right),
\end{aligned}$$

where I_{INT}, S_D, I_{AN}, D_{AN}, I_{FR}, D_{FR}, I_{RS}, and D_{RS} are the average size of Request, Response, Attachment Notification, Attachment Notification Confirmation, Face Remove, Face Remove Confirmation, Re-sync Request, and Re-sync Response messages, respectively.

Proof. The contribution to the communication overhead due to the Subscription, Data Exchange and Re-synchronization procedures remains the same as for the scenario with multiple consumers. Differently, the overhead due to the Attachment and Inspection procedure must be revised as already done in Theorem 2. Therefore, by substituting (3.3), (3.5), (3.6), (3.8), (3.10), (3.13), (3.14) in (3.1), the theorem is proved. □

3.3 Numerical results

The communication overhead generated by the conceived protocol architecture is herein numerically evaluated in scenarios with different topology, mobility, and application settings, as well as various numbers of mobile consumers. The numerical evaluation also includes a comparison with the communication overhead generated by the behavior of several reference pull-based approaches, like those in [47], [48], [60]–[62], [78]–[81], that do not solve the problems related to stale paths and loss of synchronization. This would provide a clear idea about the ability of the proposed approach to offer evident benefits in a very large set of (non-simplistic) assumptions. MATLAB scripts are used to simulate consumer mobility and obtain numerical results[1]. The conducted numerical study considers real topologies (and, in turn, real communication paths and stale disjoint paths) generated through a well-known simulator, capable of iteratively generating scale-free topologies, namely Representative Internet Topology gEnerator (BRITE) [90]. At the same time, results have been obtained by considering real application models (from the state of the art [82], [84], [91]–[93] and 3GPP standards [94]). Regarding the size of the messages (including Interest and Data packets), the 0.1.1 version of the NDN packet format specification, as described in [95], is taken into account. In this way, the considerations and obtained results become comparable with those reported in other recent works published in scientific literature [3], [12], [31], [96], [97]. The list of investigated KPIs are defined below:

- **Average communication overhead on the data plane**. According to the discussion presented in both Section 3.1 and Section 3.2, it represents the amount of unuseful data delivered across stale disjoint links. The proposed study compares the behavior of the conceived protocol architecture with respect to the pure NDN deployment where consumer mobility is simply addressed through the baseline pull-based strategy (as analytically formulated in [12]).
- **Average communication overhead on the control plane**. It quantifies the impact of control messages exchanged within the network, generated by the Attachment and the Inspection procedures. First, the overhead generated with the implementation of Neighbor and Router Inspection procedures is compared by assuming a control plane fully

---

[1] The code is available on GitHub at https://github.com/telematics-lab/SDN-MEC-ICN-consumer-mobility.

implemented through information-centric primitives (which represents the proposed solution that makes use of POF instructions to deliver control messages). To provide further insight, the discussed investigation also considers the average communication overhead on the control plane achieved when both the Inspection procedures are implemented through the OpenFlow protocol. Of course, POF-based and OpenFlow-based implementations of the Inspection procedures follow the same logic. The contribution to the communication overhead introduced by the implementation based on the conventional OpenFlow protocol of the Inspection procedure is partially revised as summarized in what follows. The Attachment procedure comprises a Port Status message, to notify the attachment, and a Packet_IN message, to notify the SDC about the topic-name of interest. The Inspection procedure considers a Modify Flow Entry message to remove the face related to the consumer from aggregated PIT entries, a Modify Flow Entry message to delete PIT entries related to the consumer only, and a Flow Removed message to notify the SDC about the deleted entries and enable the Interest path discovery.

- **Overhead reduction**. It represents the difference, expressed in percentage, between the overhead generated by the baseline pull-based approach implemented in a pure NDN approach and the overhead generated by the proposed protocol architecture. The study considers both the POF-based and the OpenFlow-based implementations for the control plane, as well as the Router Inspection and Neighbor Inspection procedures. The overhead reduction is calculated as:

$$O_{reduction} = \frac{\bar{O}\Big|_{\text{Baseline NDN}} - \bar{O}}{\bar{O}\Big|_{\text{Baseline NDN}}} 100,$$

where $\bar{O}\Big|_{\text{Baseline NDN}}$ is equal to $\frac{1}{\Delta t_c}\left(S_R + \left(1 + \frac{\Delta t_c}{T_D}\right) d I_{INT} + (A-1) S_D\right)$, as presented in [12].

In scenarios with multiple consumers, the overhead generated by the baseline pull-based approach implemented in a pure NDN approach, $\bar{O}\Big|_{\text{Baseline NDN}}$, is partially revised by taking into account the considerations illustrated in Th. 3. In this case, $\bar{O}\Big|_{\text{Baseline NDN}}$ is equal to $\frac{L}{\Delta t_c}\left(S_R + \left(d + \frac{D \Delta t_c}{L T_D}\right) I_{INT} + (A-1) S_D\right)$.

- **Bandwidth savings**. It quantifies the overall bandwidth savings achieved when considering any kind of messages exchanged during the service provisioning. Indeed, during the cell residence time, the producer generates an average number of Data packets equal to $\frac{E[\Delta t_c]}{T_D}$. Given the size of a Data packet S_D, the average bandwidth consumed for actually transmitting new versions of requested contents during the average cell residence time, and for all the links of the path established between the producer and the MEC entity serving the mobile consumer, is equal to: $E\left[\left(\frac{E[\Delta t_c]}{T_D} S_D d_{M \to P}\right) / E[\Delta t_c]\right] = \frac{dS_D}{T_D}$. Therefore, the overall bandwidth savings, $B_{savings}$ in a scenario with a single consumer is calculated as:

$$
\begin{aligned}
B_{savings} &= \frac{\left(\bar{O}\Big|_{\text{Baseline NDN}} + \frac{dS_D}{T_D}\right) - \left(\bar{O} + \frac{dS_D}{T_D}\right)}{\bar{O}\Big|_{\text{Baseline NDN}} + \frac{dS_D}{T_D}} 100 = \\
&= \frac{\bar{O}\Big|_{\text{Baseline NDN}} - \bar{O}}{\bar{O}\Big|_{\text{Baseline NDN}} + \frac{dS_D}{T_D}} 100.
\end{aligned}
$$

In scenarios with multiple consumers, the overall bandwidth savings, $B_{savings}$, is partially revised, by taking into account the considerations illustrated in Th. 3, as reported below:

$$
\begin{aligned}
B_{savings} &= \frac{\left(\bar{O}\Big|_{\text{Baseline NDN}} + \frac{DS_D}{T_D}\right) - \left(\bar{O} + \frac{DS_D}{T_D}\right)}{\bar{O}\Big|_{\text{Baseline NDN}} + \frac{DS_D}{T_D}} 100 = \\
&= \frac{\bar{O}\Big|_{\text{Baseline NDN}} - \bar{O}}{\bar{O}\Big|_{\text{Baseline NDN}} + \frac{DS_D}{T_D}} 100.
\end{aligned}
$$

- **Memory saving**. The memory saving is the upper bound of the amount of memory spared from PIT and CS tables by the proposed protocol for all routers of the stale disjoint path. In fact, as the Inspection procedure removes stale information from the routers of the stale disjoint path, it frees up precious space in PIT tables and spares CS from storing unuseful data. According to the discussion presented in both Section 3.1 and Section 3.2, the memory saving with respect to the baseline NDN approach is equal to:

$$M_{savings} = (A-1)S_{PIT} + (A-1)S_D, \tag{3.16}$$

where A, S_{PIT}, and S_D are the average number of routers on the stale disjoint path, the average size of a PIT entry, and the average size of a Data packet, respectively

3.3.1 Main parameters settings and topology models

The conducted study considers N network attachment points randomly distributed within a 10 km x 10 km urban area, corresponding to a medium-sized city. The distributed routers compose a multi-hop wireless mesh network. The average cell radius r is set to 50 m and 150 m [12], [98]. These values translate to a number of routers in the considered urban area N equal to 12732 and 1415, respectively, following $N = Area/(\pi r^2)$. On the other hand, the consumer speed is set to 3 or 30 km/h [12], [98]. Both the cell radius and consumer speed values refer to urban scenarios [98]. The average round trip time τ for wireless mesh networks is set to 0.1 s [99].

Regarding the application model, a wide range of combinations of the average application payload size is chosen between 5 kB and 50 MB and the average time interval between the generation of two consecutive contents in the range between 0.1 s and 10^4 s are taken into account. Accordingly, the resulting analysis is useful to describe the behavior of the proposed approach for many emerging applications, including HTTP [82], Big Data transfer [91], [92], aggregated sensor data [100], remote control of mobile robots [94], and adaptive video streaming [84], [93].

The size of the Subscription Request message is calculated as in [101]. The size of the exchanged messages for unuseful Data and control packets is calculated by considering the structure of Interest and Data packets [3], [95], [102] (see Table 3.3), an average size for the content name equal to 17.44 B [103], and the average size of the application payload, as mentioned before. The size of OpenFlow messages is calculated by considering the OpenFlow Switch Specification 1.5.1 [104].

Considering that the communication overhead also depends on the length of the path established between network node pairs, modeling the network topology becomes of paramount importance. This work considers a wireless, multi-hop, mesh, scale-free network topology with N routers, where the 1-hop neighbors are distributed according to a power law with

TABLE 3.3: Structure and average size of both Interest and Data, according to NDN specifications [3], [31], [95]–[97].

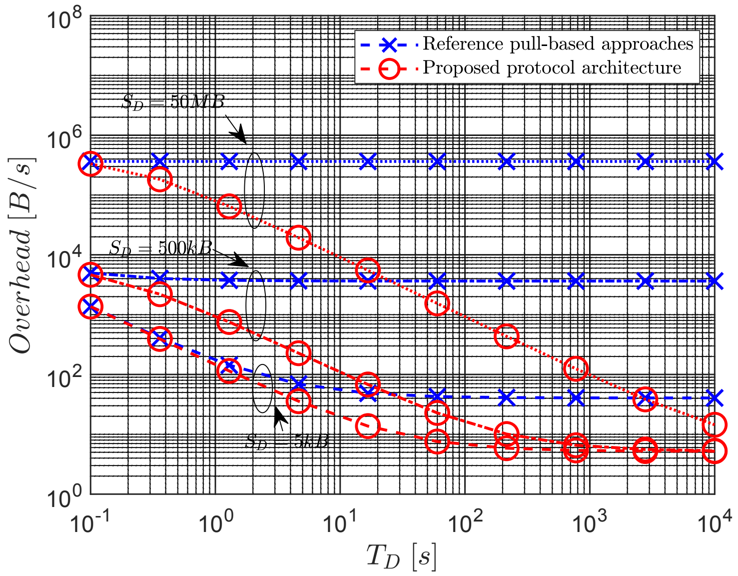
Interest Field	**Size [B]**
Nonce	4
Scope	1
Nack Type	1
InterestLifetime	2
Name	2 + 17.44
Selectors	2
Options	2

Data Field	**Size [B]**
Name	2 + 17.44
Content	2 + application payload size (5 kB to 50 MB)
Signature	33

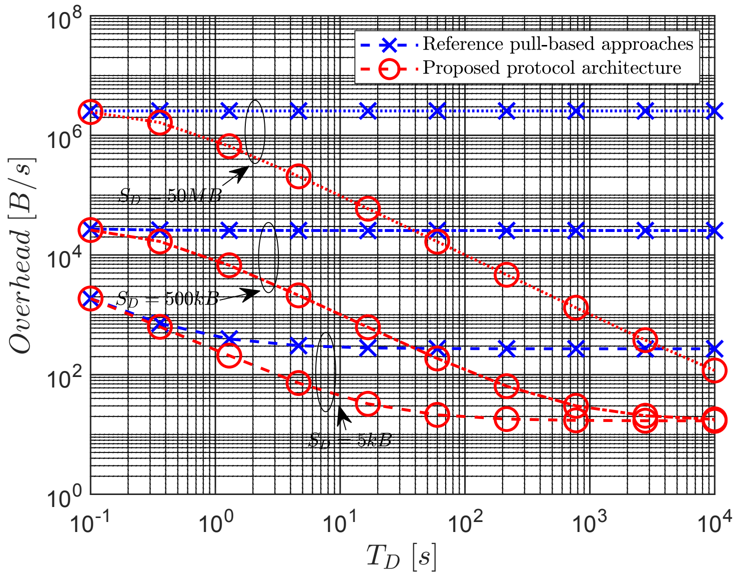
factor $\gamma = 3$ [87], [105]. The average number of neighbors per node k is equal to $k = 2m$, where m is the number of neighbors a new node is attached to according to the preferential attachment law of the Barabasi-Albert model [87]. Different network topologies have been generated with the BRITE simulator [90]. Moreover, without loss of generality, it is assumed that each of these routers represents a network attachment point and hosts a MEC entity. The SDC is randomly attached to one of the available routers. The geo-referenced topologies generated with BRITE have been imported in MATLAB for evaluating the size of the disjoint path. Here, the position of both consumers and producer are randomly chosen at the beginning. Then, the consumers are enforced to move within the considered urban area according to the random direction mobility model. During the simulation, the consumers attach to the closest network attachment points and, therefore, trigger several handover procedures. The shortest path between any couple of nodes is calculated according to the Dijkstra algorithm. In this context, computer simulations are carried out to estimate the average number of routers on the stale disjoint path, that is A, and to verify that the average shortest path length between any nodes pair, d, is equal to $d = logN/log(logN)$ (according to [87], referring to a scale-free network topology). For each scenario, 300 different topologies are evaluated. For each topology, 100 initial positions of consumers, producer, and SDC are considered as well. Indeed, obtained results are averaged on $3 \cdot 10^4$ realizations.

3.3.2 Numerical results for a scenario with a single consumer.

Figure 3.7 reports the average communication overhead expected on the data plane, as a function of network size, consumer speed, and application settings. As a preliminary consideration, it is possible to observe that the average communication overhead registered by the proposed protocol architecture on the data plane decreases with T_D.

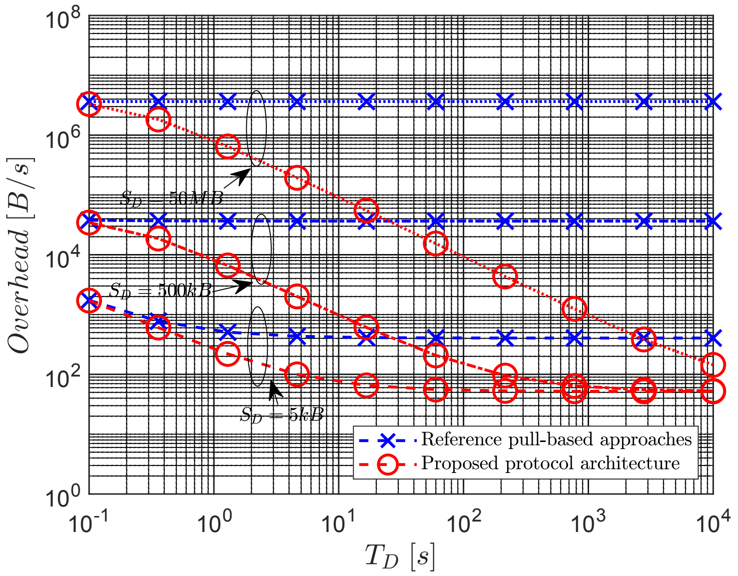


(A) $r = 150\ m, v = 3\ km/h$

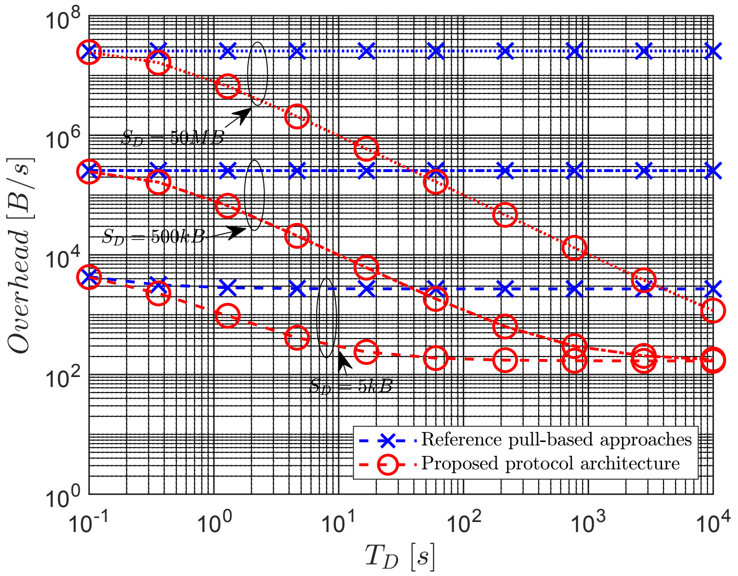


(B) $r = 50\ m, v = 3\ km/h$

FIGURE 3.7: Average communication overhead on the data plane, as a function of network size, consumer speed, and application settings.



(C) $r = 150\ m, v = 30\ km/h$

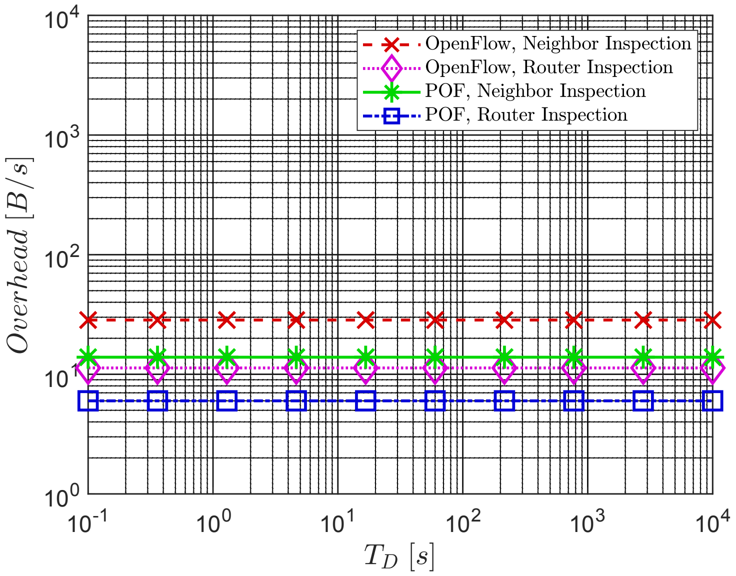


(D) $r = 50\ m, v = 30\ km/h$

FIGURE 3.7: Average communication overhead on the data plane, as a function of network size, consumer speed, and application settings.

In fact, as the time interval between the generation of two consecutive contents increases, the SDC can clean as much wrong forwarding information as possible, thus breaking down unuseful Data dissemination. The average size of the Data packet influences the communication overhead as well. The higher the application payload size, S_D, the higher the amount of unuseful Data exchanged across the stale disjoint path. Note that the average communication overhead on the data plane increases with the consumer speed and when the average cell radius decreases. In both cases, handover episodes occur more frequently, thus augmenting the number of messages exchanged during the Re-Synchronization procedure and the amount of Data delivered across an even more number of stale links. Anyway, the most important comment emerging from the curves reported in Figure 3.7 is that the proposed protocol architecture is able to always reduce the communication overhead on the data plane with respect to any other approach that exploits the pull-based approach for addressing consumer mobility in ICN deployments.

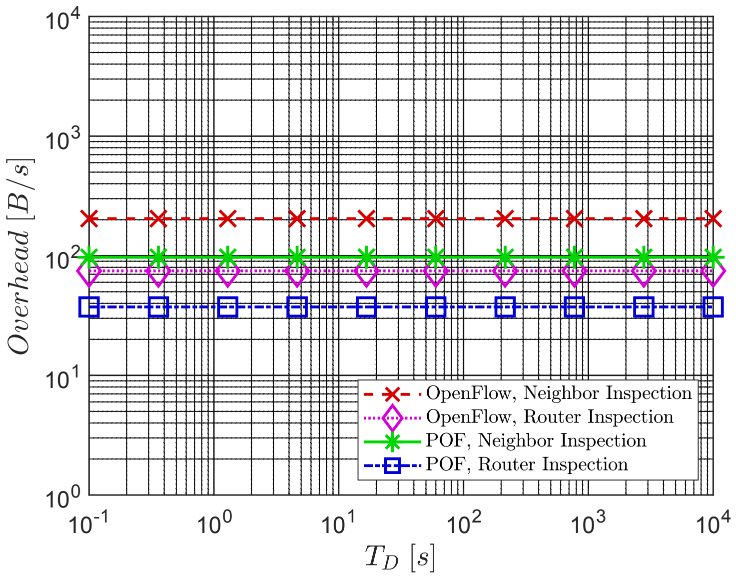
The average communication overhead generated by the proposed protocol architecture on the control plane is depicted in Figure 3.8.



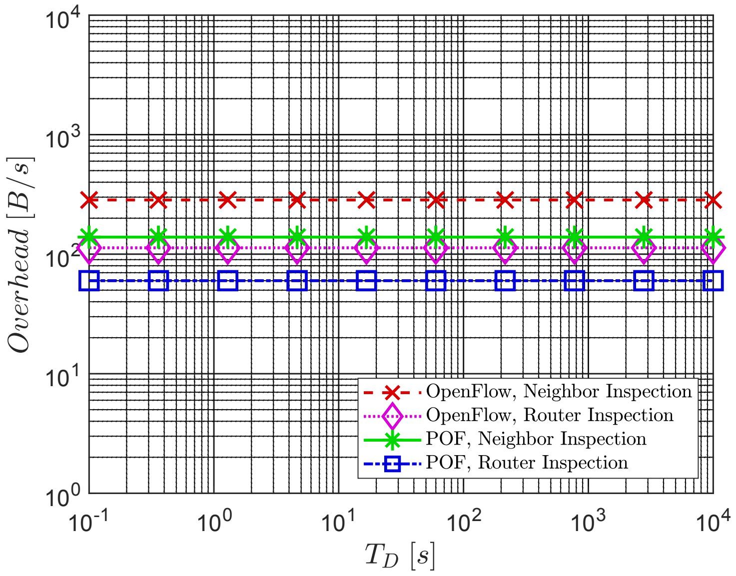
(A) $r = 150\ m, v = 3\ km/h$

FIGURE 3.8: Average communication overhead on the control plane, as a function of network size, consumer speed, and application settings.

Obtained results identify the most performant Inspection procedure (i.e., Router Inspection against Neighbor Inspection) and its most effective implementation (i.e., OpenFlow-based against POF-based). The conducted study remarks that the average communication overhead due to control messages increases with the consumer speed or when the average cell radius decreases. In both cases, in fact, the mobile consumer changes the network attachment point more frequently and triggers more times the execution of the different Attachment and Inspection procedures. Thus, a higher number of control messages are exchanged in a unit of time. At the same time, it is possible to observe that the average communication overhead does not change with T_D. The control messages of these procedures are exchanged only once during



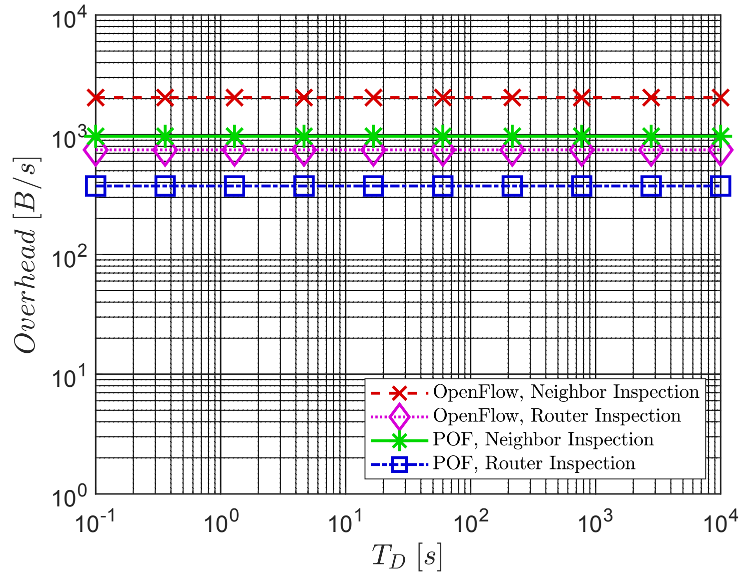
(B) $r = 50\ m, v = 3\ km/h$



(C) $r = 150\ m, v = 30\ km/h$

FIGURE 3.8: Average communication overhead on the control plane, as a function of network size, consumer speed, and application settings.

a cell residence time, hence independently from T_D. Regarding the Inspection procedure, the Router Inspection scheme ensures a lower impact on the communication overhead, because of its ability to directly inspect the routers belonging to the stale disjoint path. On the other hand, the POF-based implementation of the control plane always achieves better performance and an overhead reduction ranging from 41.17% to 51.13% with respect to an implementation based on the conventional OpenFlow protocol. This clearly

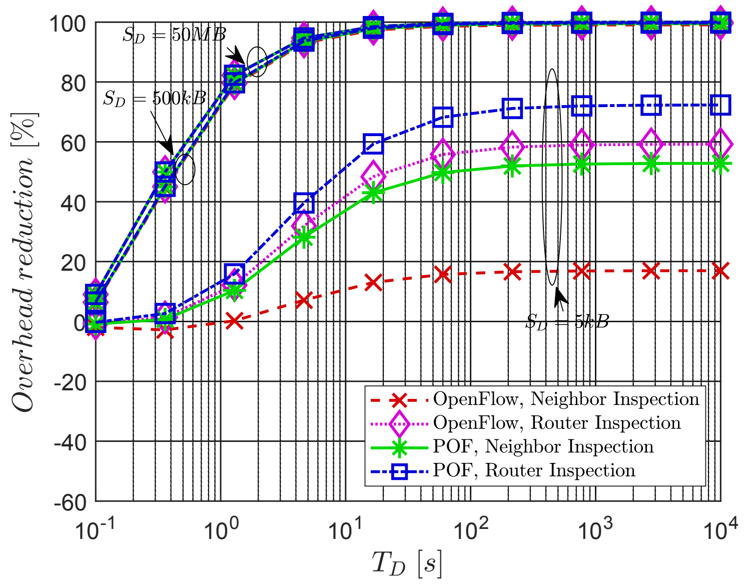


(D) $r = 50\ m, v = 30\ km/h$

FIGURE 3.8: Average communication overhead on the control plane, as a function of network size, consumer speed, and application settings.

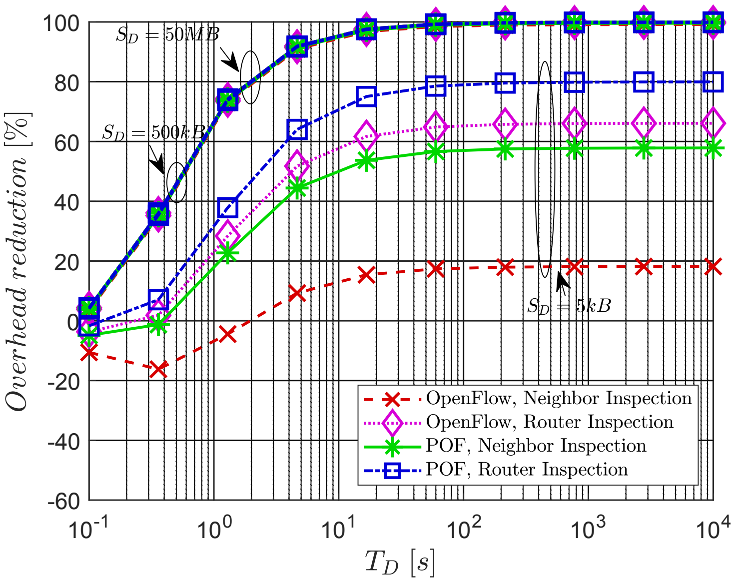
demonstrates the evident benefits offered by a control plane fully implemented through information-centric primitives.

The total overhead reduction is reported in Figure 3.9.

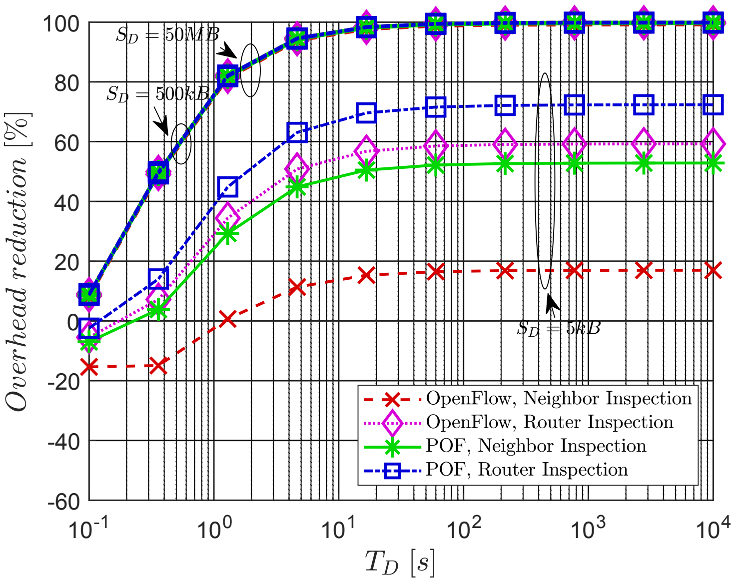


(A) $r = 150\ m, v = 3\ km/h$

FIGURE 3.9: Reduction of the average communication overhead, as a function of network size, consumer speed, and application settings.



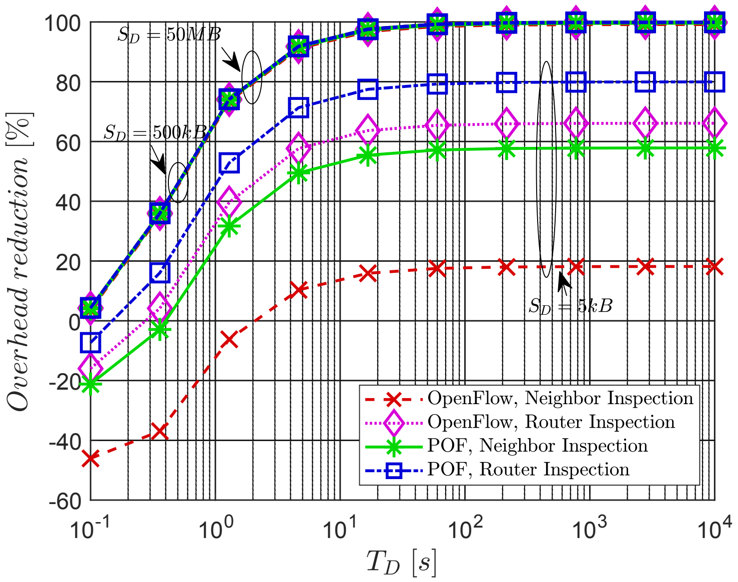
(B) $r = 50\ m$, $v = 3\ km/h$



(C) $r = 150\ m$, $v = 30\ km/h$

FIGURE 3.9: Reduction of the average communication overhead, as a function of network size, consumer speed, and application settings.

Results fully confirm the unique ability of the proposed protocol architecture to reduce the communication overhead in all the considered scenarios. For applications with limited packet size, the contribution due to control messages becomes more significant against the effect of the dissemination of unuseful Data. Indeed, regarding the scenario with $S_D = 5kB$, reported curves show that the POF-based implementation of the Router Inspection

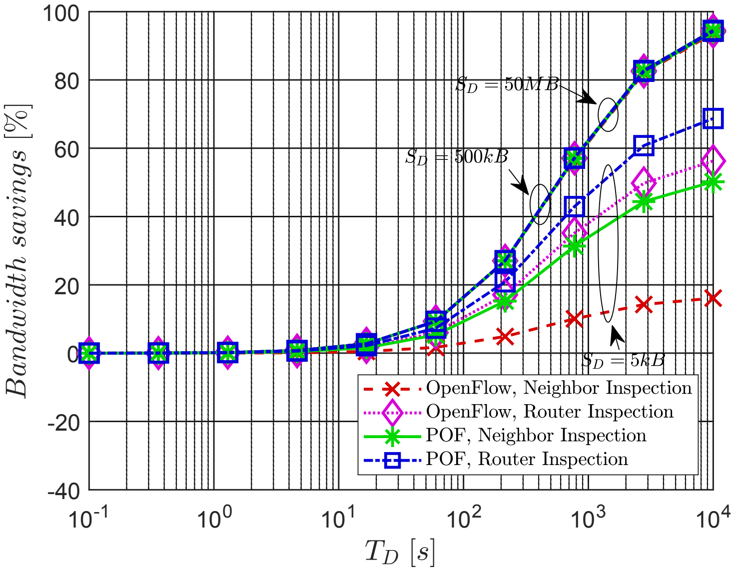


(D) $r = 50\ m$, $v = 30\ km/h$

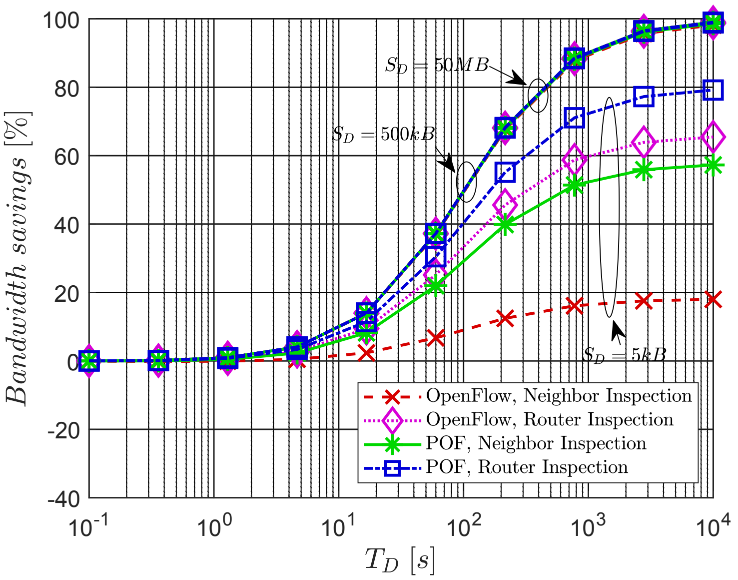
FIGURE 3.9: Reduction of the average communication overhead, as a function of network size, consumer speed, and application settings.

procedure always achieves better performance with respect to other implementations of the control plane. In other cases, an overhead reduction is almost due to the behavior of the data plane. Specifically, a reduction of up to 99.99% is achieved in those scenarios where the SDC is able to minimize (or at most erase) the waste of bandwidth due to the dissemination of Data packets across the links belonging to the stale disjoint path (i.e., when T_D and S_D increase, or the cell radius r decreases, or consumer speed v increases).

Figure 3.10 shows the overall bandwidth savings. Once again, results demonstrate the great performance gain offered by the protocol architecture presented in this work. Specifically, bandwidth savings up to 99.9% are registered in those scenarios where the SDC can minimize (or at most erase) the waste of bandwidth due to the dissemination of Data packets across the links belonging to the stale disjoint path (i.e., when T_D and S_D increase, or when the average cell radius r decreases and the consumer speed v increases).

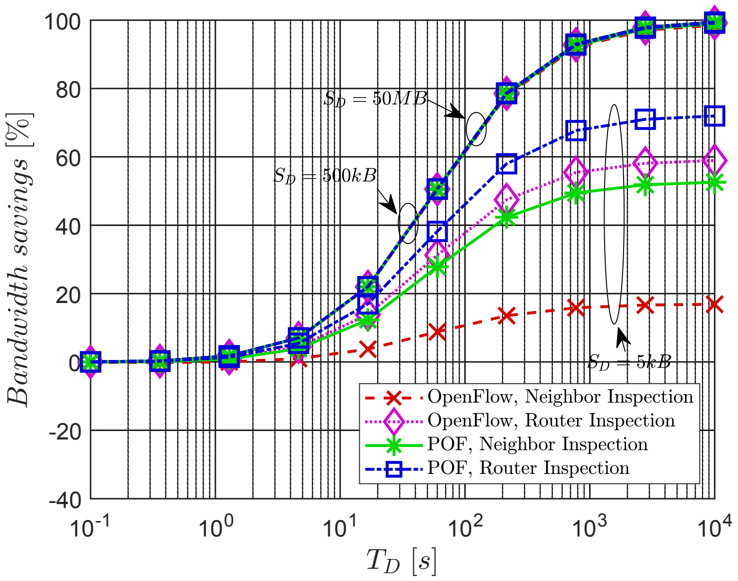


(A) $r = 150\ m, v = 3\ km/h$

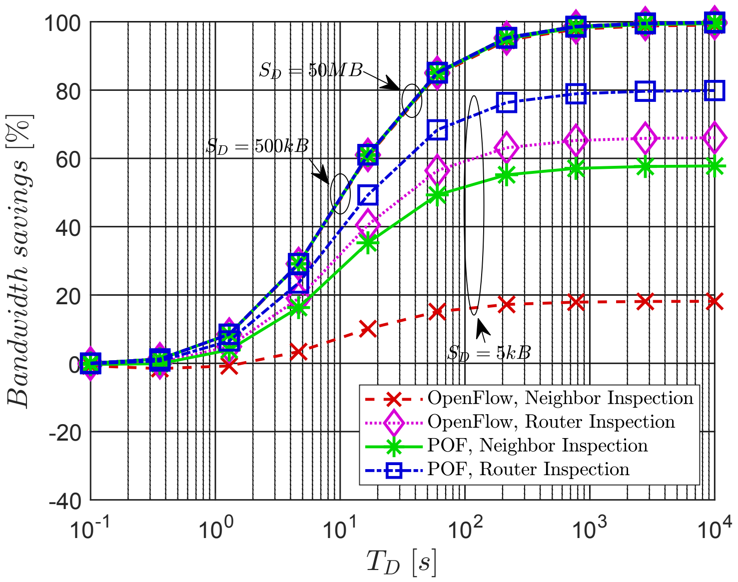


(B) $r = 50\ m, v = 3\ km/h$

FIGURE 3.10: Bandwidth savings, as a function of network size, consumer speed, and application settings.



(C) $r = 150\ m, v = 30\ km/h$



(D) $r = 50\ m, v = 30\ km/h$

FIGURE 3.10: Bandwidth savings, as a function of network size, consumer speed, and application settings.

Finally, Table 3.4 reports the memory savings offered by the proposed protocol architecture, calculated by considering an average size of PIT entries, that is S_{PIT}, equal to 275 B [106]. Surely, the memory savings increase with the average size of Data packets, S_D. At the same time, it is influenced by the average cell radius r. It is already explained that an increment of the average cell radius brings to lower network size. When the network size increases, the length of the stale disjoint path increases as well. Here, the number of PIT entries erased by the Inspection procedure increases as well.

TABLE 3.4: Total amount of memory savings in both CS and PIT.

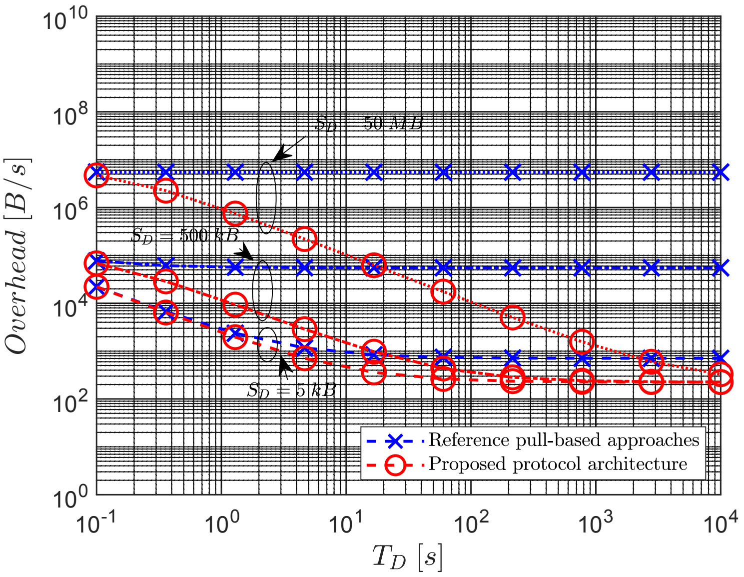
		S_D		
		5 kB	500 kB	50 MB
Average cell radius	r= 50 m	25.5 kB	2.42 MB	242 MB
	r= 150 m	10.9 kB	1.03 MB	103 MB

3.3.3 Numerical results for a scenario with multiple consumers.

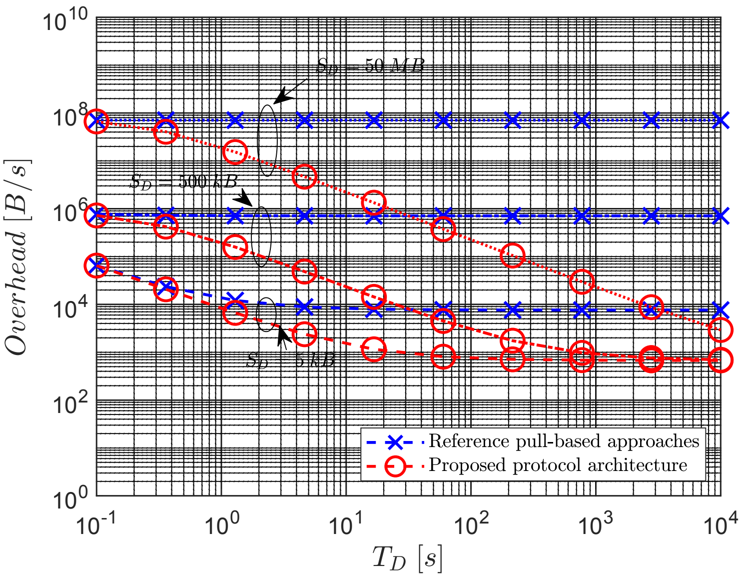
This Section describes the behavior of the proposed protocol architecture in scenarios with multiple consumers. First of all, Figure 3.11 and Figure 3.12 report the average communication overhead expected on the data plane, when the number of mobile consumers is equal to 40 and 160, respectively.

In line with all the comments discussed before, it is possible to observe that the average communication overhead on the data plane decreases when the time interval between the generation of two consecutive real-time contents T_D increases, the cell radius r increases, the consumer speed v decreases, and the application payload size S_D decreases.

The number of mobile consumers affects the average communication overhead expected on the data plane as well. In fact, the higher number of mobile consumers, the higher number of handover episodes managed within the multi-hop wireless mesh network. This, in turn, generates a greater amount of stale disjoint links and a consequent increment of the communication overhead due to both Re-Synchronization and Data Exchange procedures.

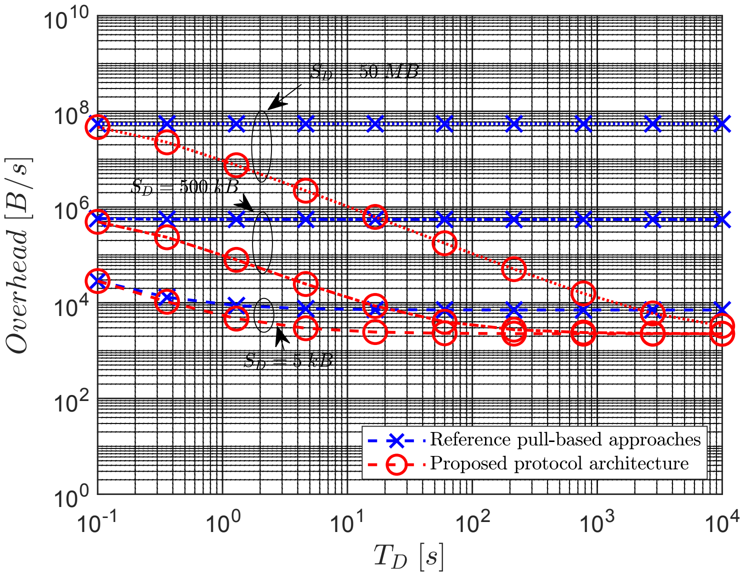


(A) $r = 150\ m, v = 3\ km/h$

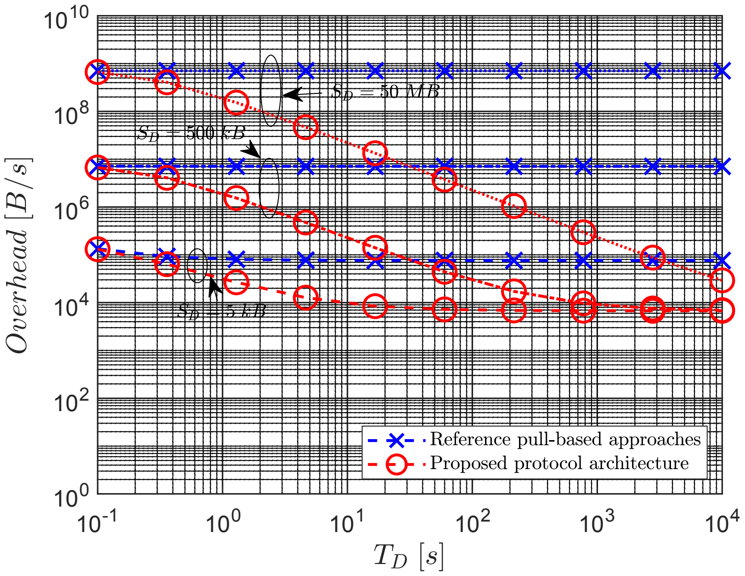


(B) $r = 50\ m, v = 3\ km/h$

FIGURE 3.11: Average communication overhead on the data plane with 40 mobile consumers, as a function of network size, consumer speed, and application settings.

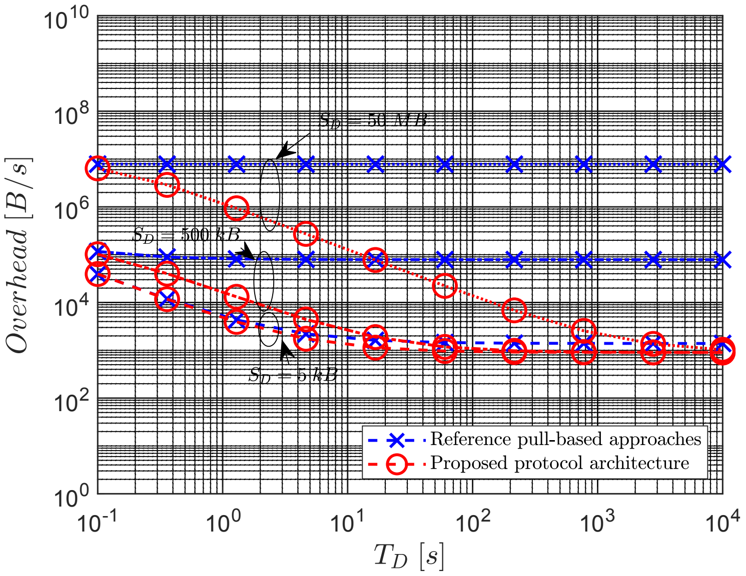


(C) $r = 150\ m, v = 30\ km/h$

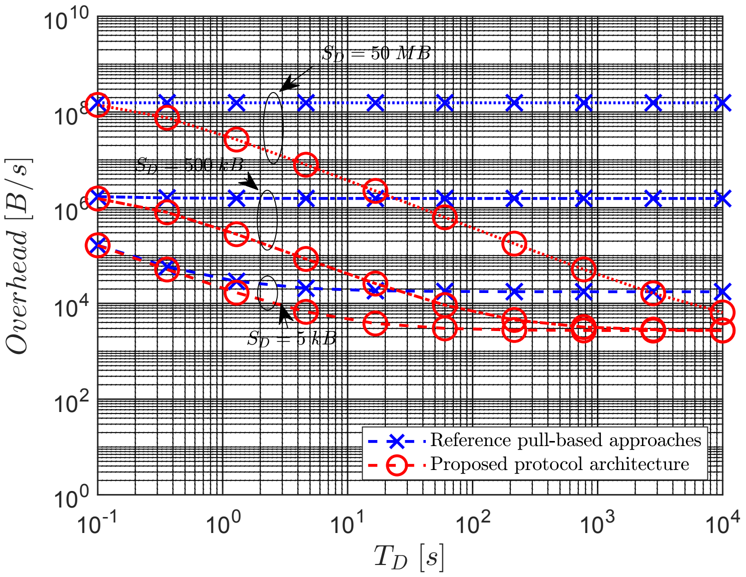


(D) $r = 50\ m, v = 30\ km/h$

FIGURE 3.11: Average communication overhead on the data plane with 40 mobile consumers, as a function of network size, consumer speed, and application settings.

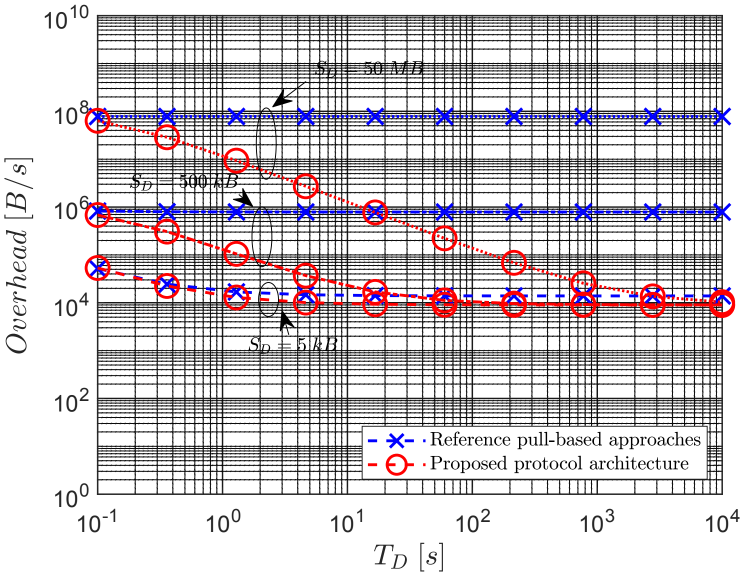


(A) $r = 150\ m, v = 3\ km/h$

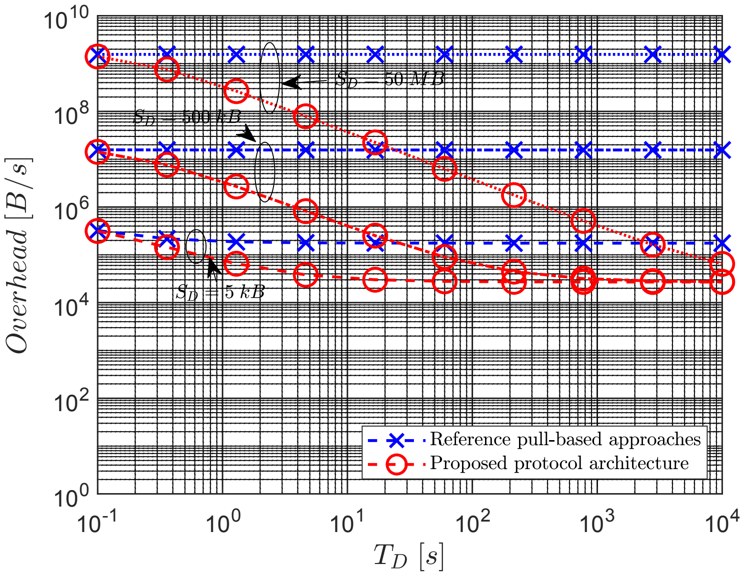


(B) $r = 50\ m, v = 3\ km/h$

FIGURE 3.12: Average communication overhead on the data plane with 160 mobile consumers, as a function of network size, consumer speed, and application settings.



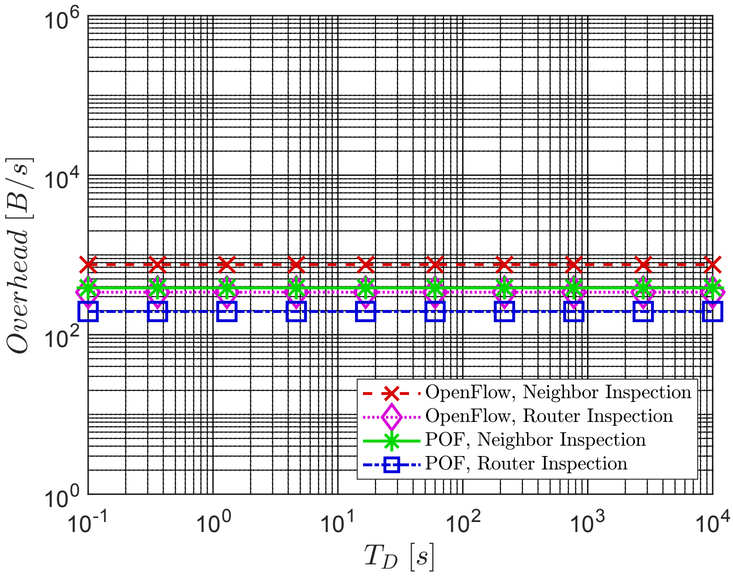
(C) $r = 150m, v = 30km/h$



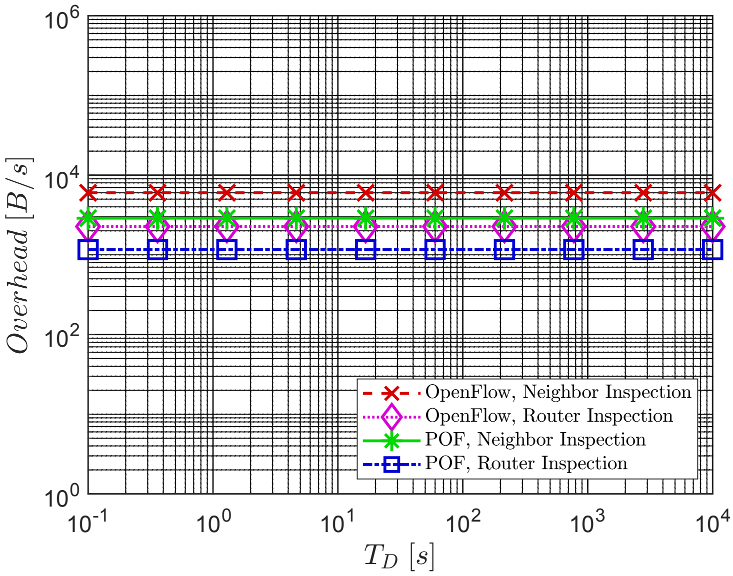
(D) $r = 50m, v = 30km/h$

FIGURE 3.12: Average communication overhead on the data plane with 160 mobile consumers, as a function of network size, consumer speed, and application settings.

The same considerations can be formulated by observing the average communication overhead expected on the control plane, as reported in Figure 3.13 (for the scenario with 40 mobile consumers) and Figure 3.14 (for the scenario with 160 mobile consumers). Therefore, the average communication overhead on the control plane decreases when the cell radius r increases, the consumer speed v decreases, and the application payload size S_D decreases.

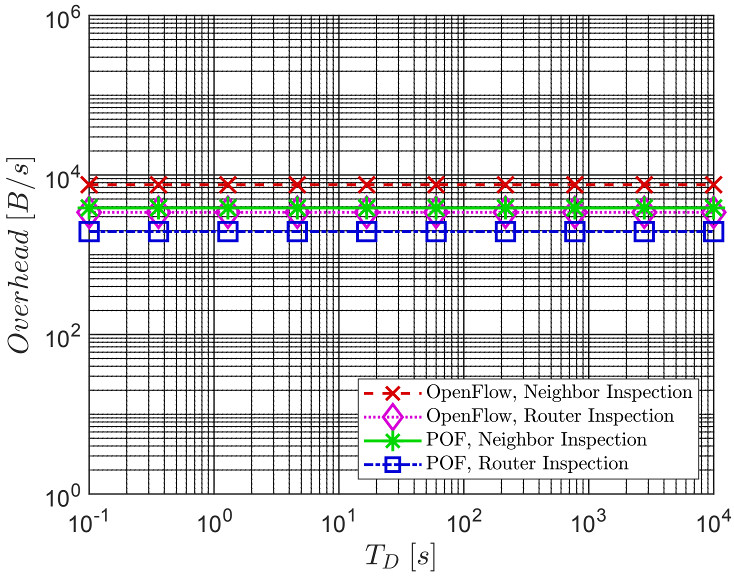


(A) $r = 150\ m, v = 3\ km/h$

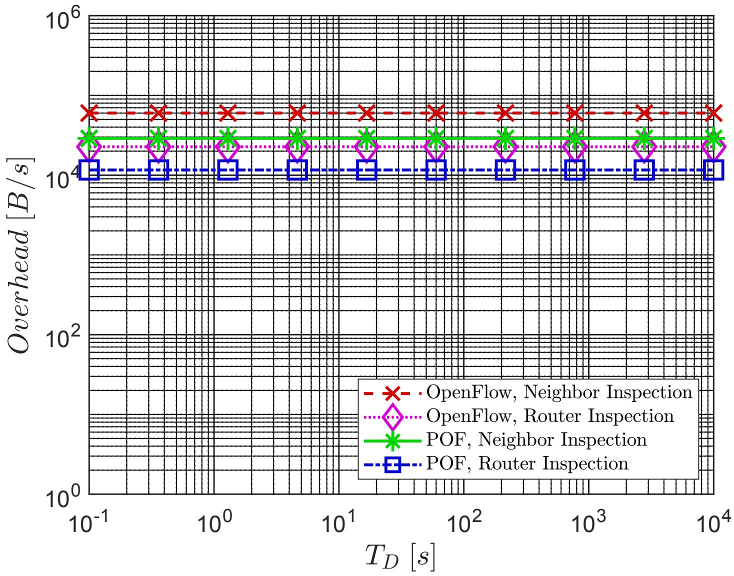


(B) $r = 50\ m, v = 3\ km/h$

FIGURE 3.13: Average communication overhead on the control plane with 40 mobile consumers, as a function of network size, consumer speed, and application settings.



(C) $r = 150\ m, v = 30\ km/h$

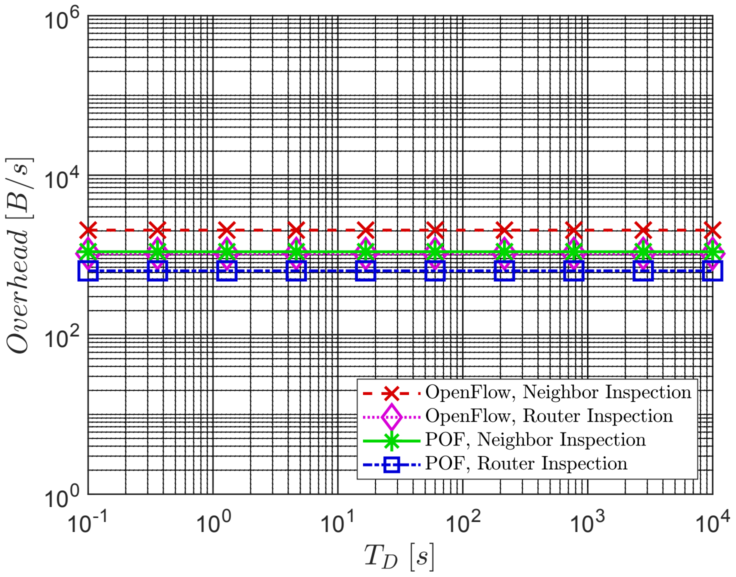


(D) $r = 50\ m, v = 30\ km/h$

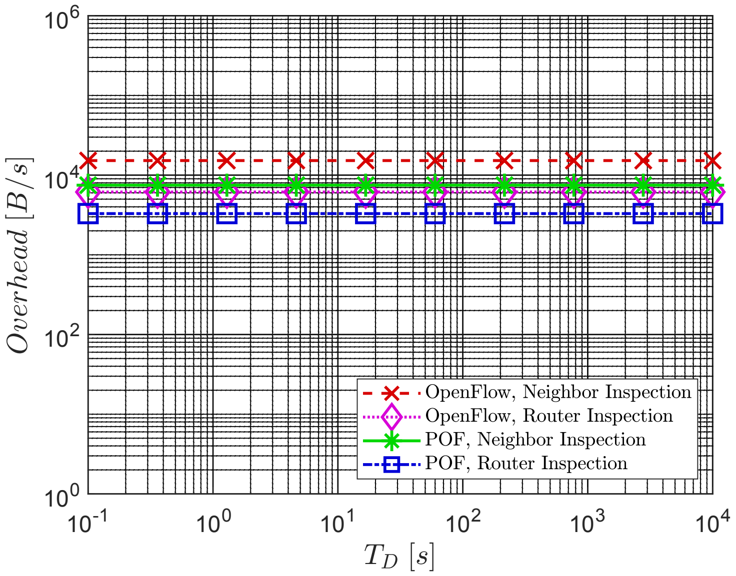
FIGURE 3.13: Average communication overhead on the control plane with 40 mobile consumers, as a function of network size, consumer speed, and application settings.

Similarly to the scenario with a single consumer, the average communication overhead on the control plane does not change with T_D, as the procedures are executed only once during a handover. In addition, as for the data plane, the higher number of mobile consumers, the higher number of messages generated by Attachment and Inspection procedures and delivered through the control plane.

Nevertheless, it is very important to remark that, even in the case of multiple consumers, the POF-based implementation of the control plane still registers an evident overhead reduction, which ranges from 29.36% to 50.16% with respect to the implementation based on the conventional OpenFlow protocol. Once again, this demonstrates the effectiveness of a control plane fully implemented through information-centric primitives.

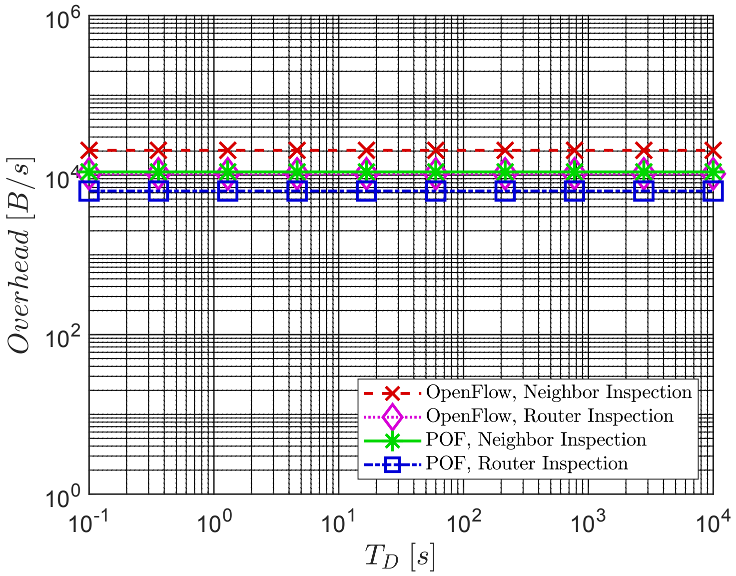


(A) $r = 150\ m, v = 3\ km/h$

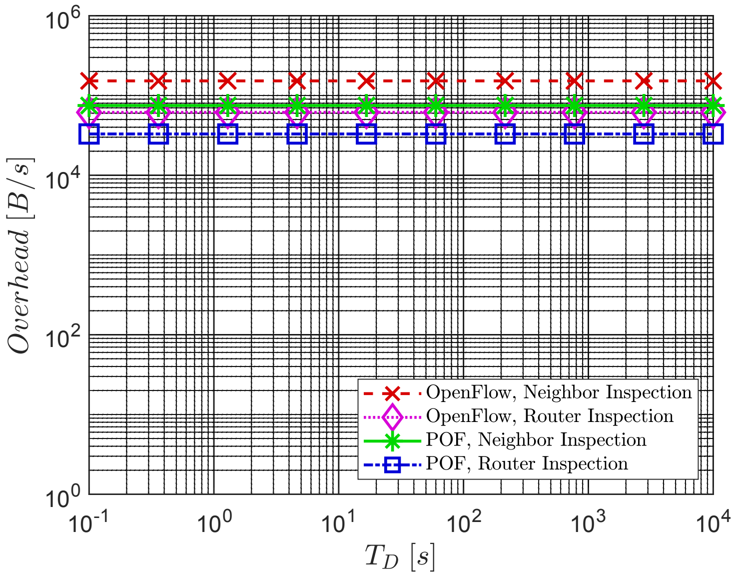


(B) $r = 50\ m, v = 3\ km/h$

FIGURE 3.14: Average communication overhead on the control plane with 160 mobile consumers, as a function of network size, consumer speed, and application settings.



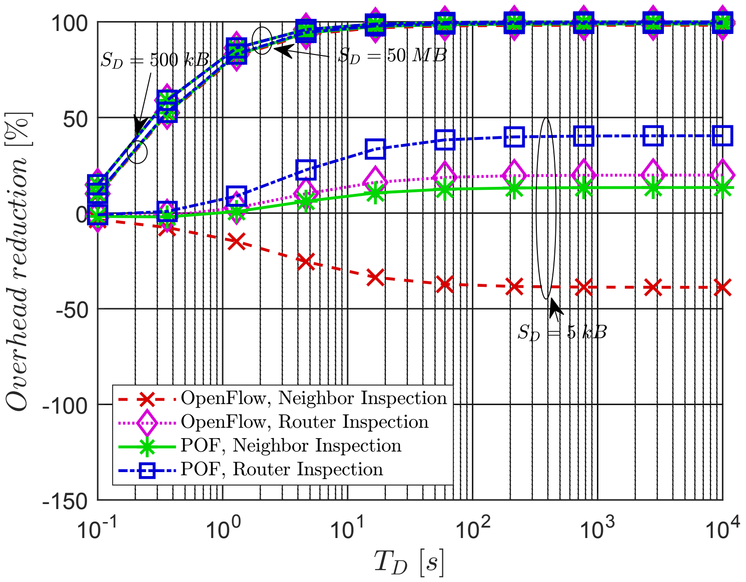
(c) $r = 150\ m, v = 30\ km/h$



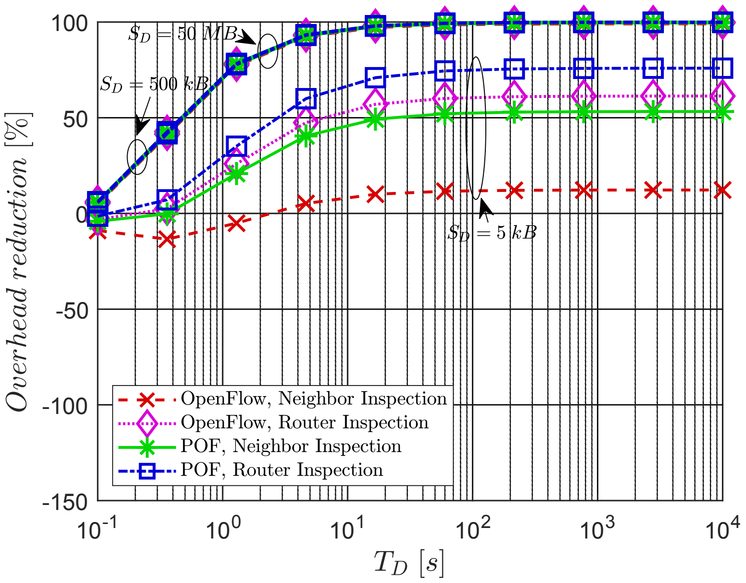
(d) $r = 50\ m, v = 30\ km/h$

FIGURE 3.14: Average communication overhead on the control plane with 160 mobile consumers, as a function of network size, consumer speed, and application settings.

Figure 3.15 and Figure 3.16 report the average overhead reduction expected for scenarios with 40 and 160 mobile consumers, respectively.



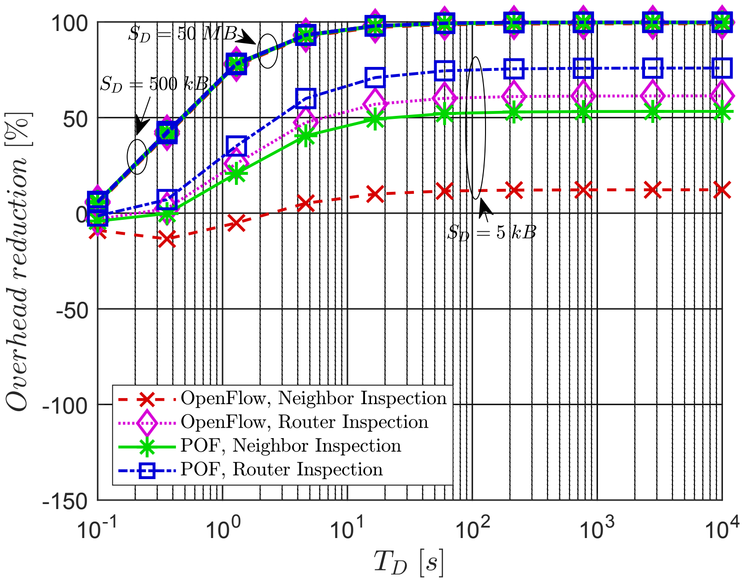
(A) $r = 150\ m, v = 3\ km/h$



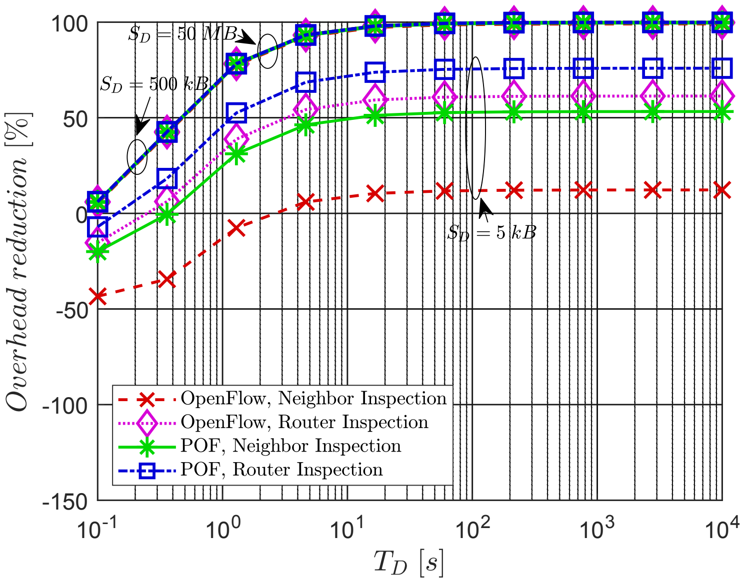
(B) $r = 50\ m, v = 3\ km/h$

FIGURE 3.15: Reduction of the average communication overhead with 40 mobile consumers, as a function of network size, consumer speed, and application settings.

The benefits highlighted in scenarios with a single consumer are still evident in most cases. When the application payload size is equal to $500kB$ and $50MB$, for example, the proposed protocol architecture ensures an overhead reduction of up to 99.99%. Moreover, the adoption of the Routing Inspection procedure and a POF-based implementation of the control plane always



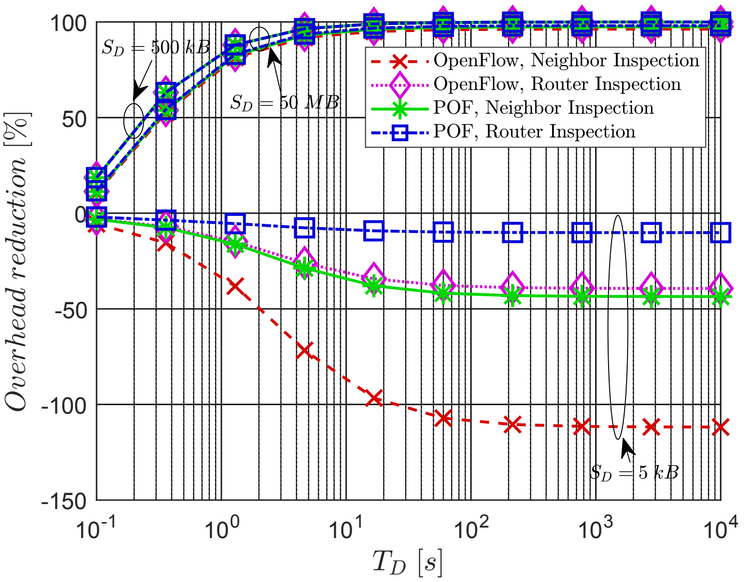
(C) $r = 150\ m, v = 30\ km/h$



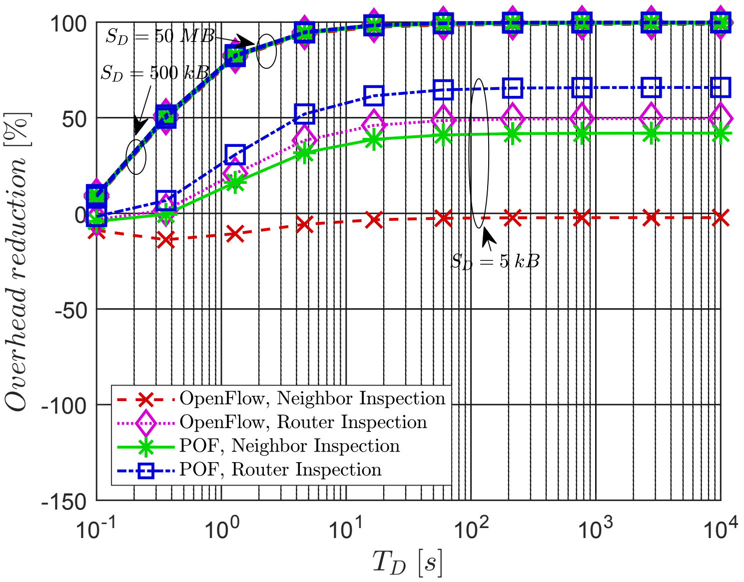
(D) $r = 50\ m, v = 30\ km/h$

FIGURE 3.15: Reduction of the average communication overhead with 40 mobile consumers, as a function of network size, consumer speed, and application settings.

guarantee the best performance. On the contrary, the results also remark that, in scenarios with lower network size, lower traffic load, and a very high number of mobile consumers asking for the same real-time contents, the amount of bandwidth consumed on the control plane tends to reach (or in some cases exceeds) the overhead saved on the data plane. However, the adoption of the Routing Inspection procedure and a POF-based implementation of the control plane emerge as suitable choices also in these extreme



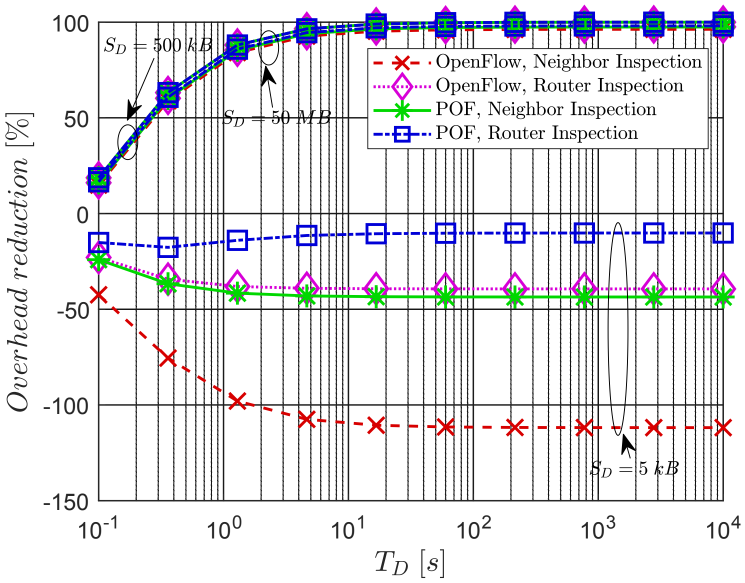
(A) $r = 150\ m, v = 3\ km/h$



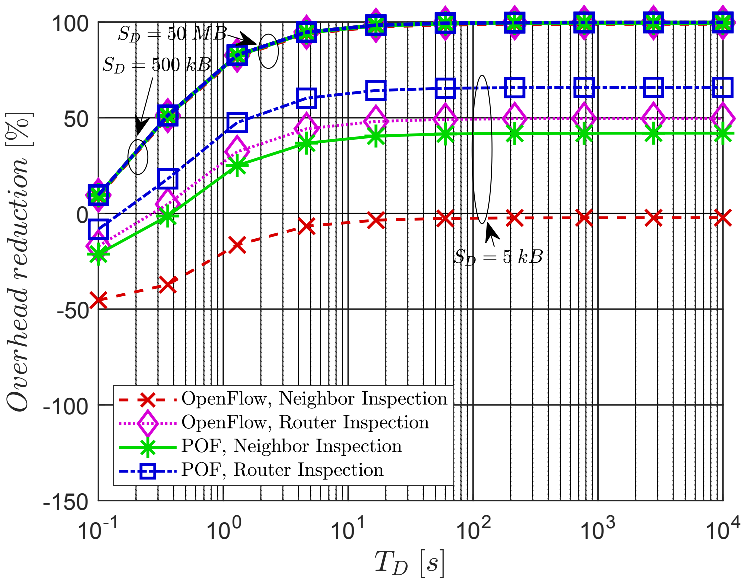
(B) $r = 50\ m, v = 3\ km/h$

FIGURE 3.16: Reduction of the average communication overhead with 160 mobile consumers, as a function of network size, consumer speed, and application settings.

conditions, while proving that the proposed protocol architecture still offers unique capabilities in the transparent and flexible management of control and communication functionalities in a multi-hop wireless network.



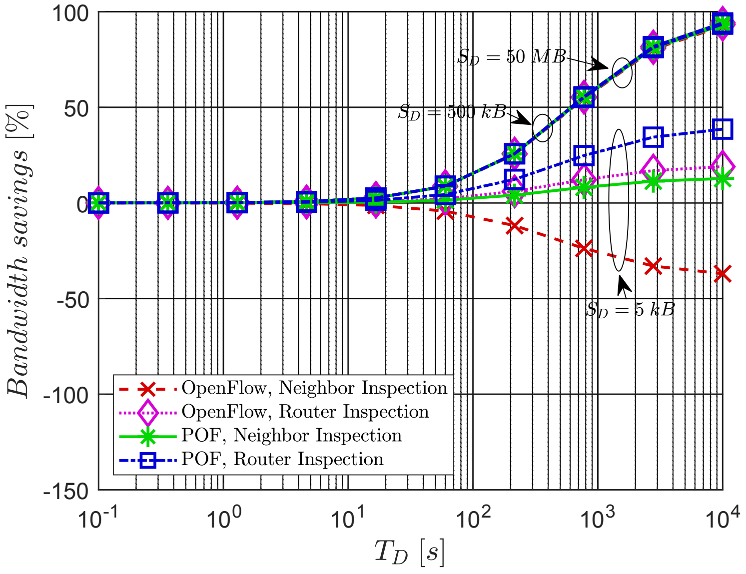
(C) $r = 150\ m, v = 30\ km/h$



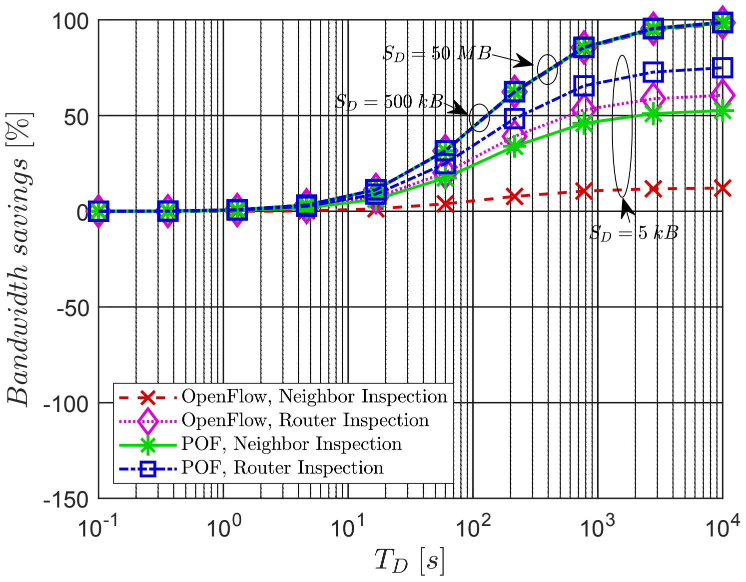
(D) $r = 50\ m, v = 30\ km/h$

FIGURE 3.16: Reduction of the average communication overhead with 160 mobile consumers, as a function of network size, consumer speed, and application settings.

The analysis of bandwidth savings reported in Figure 3.17 (for the scenario with 40 mobile consumers) and 3.18 (for the scenario with 160 mobile consumers) further confirms the previous considerations.

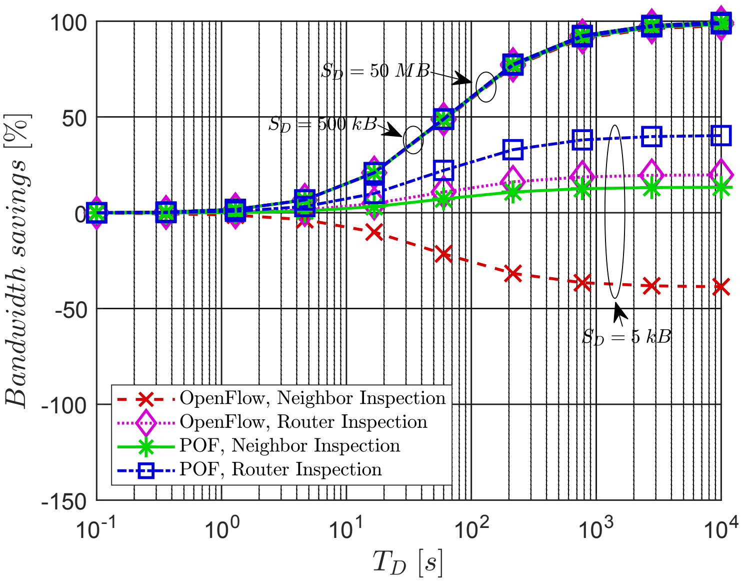


(A) $r = 150\ m, v = 3\ km/h$

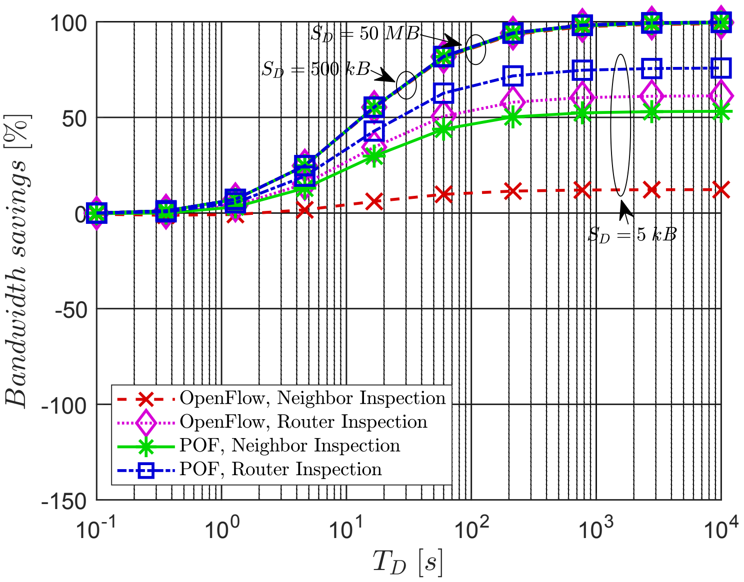


(B) $r = 50\ m, v = 3\ km/h$

FIGURE 3.17: Bandwidth savings with 40 mobile consumers, as a function of network size, consumer speed, and application settings.

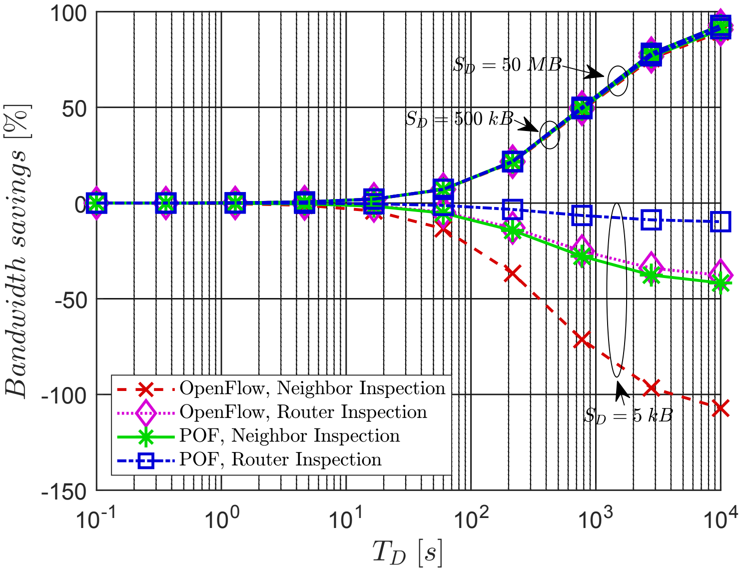


(C) $r = 150\ m, v = 30\ km/h$

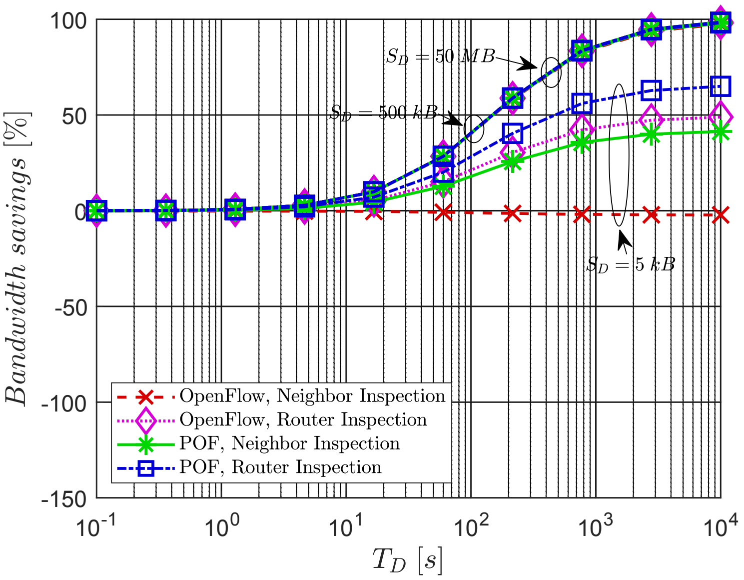


(D) $r = 50\ m, v = 30\ km/h$

FIGURE 3.17: Bandwidth savings with 40 mobile consumers, as a function of network size, consumer speed, and application settings.

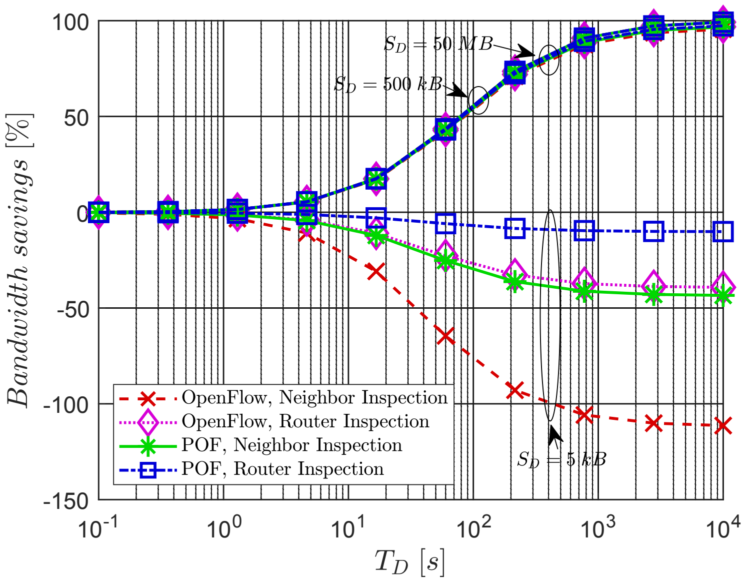


(A) $r = 150\ m, v = 3\ km/h$

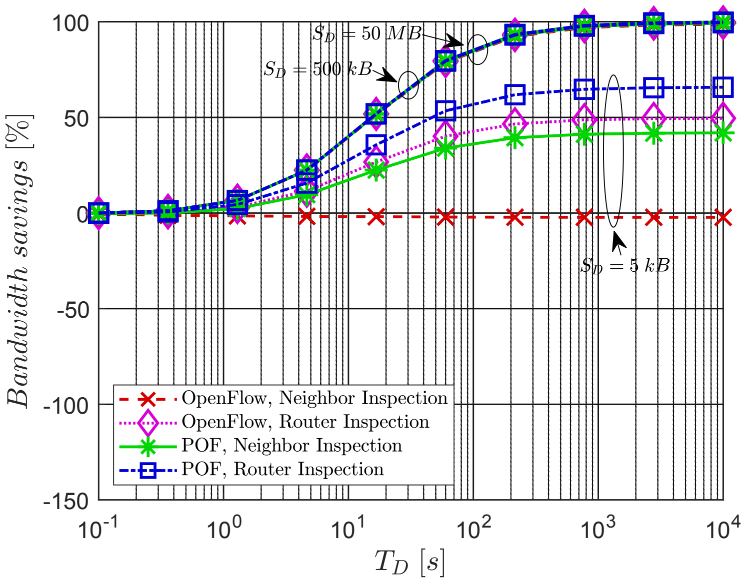


(B) $r = 50\ m, v = 3\ km/h$

FIGURE 3.18: Bandwidth savings with 160 mobile consumers, as a function of network size, consumer speed, and application settings.



(C) $r = 150\ m, v = 30\ km/h$



(D) $r = 50\ m, v = 30\ km/h$

FIGURE 3.18: Bandwidth savings with 160 mobile consumers, as a function of network size, consumer speed, and application settings.

To conclude, Table 3.5 and Table 3.6 report the memory savings obtained by the proposed protocol in a scenario with 40 and 160 mobile consumers, respectively. The results report that less memory can be spared throughout the wireless mesh network as the number of mobile consumers grows. This is due to the aggregation of requests in NDN, which intrinsically reduces the waste of memory as the number of consumers in the same network grows. But, in any case, the conceived approach still continues to offer some benefits also in this perspective.

TABLE 3.5: Total amount of memory savings in both CS and PIT with 40 mobile consumers.

		S_D		
		5 kB	500 kB	50 MB
Average cell radius	r= 50 m	17.6 kB	1.67 MB	167 MB
	r= 150 m	4.10 kB	0.389 MB	38.9 MB

TABLE 3.6: Total amount of memory savings in both CS and PIT with 160 mobile consumers.

		S_D		
		5 kB	500 kB	50 MB
Average cell radius	r= 50 m	9.68 kB	0.918 MB	91.8 MB
	r= 150 m	1.46 kB	0.138 MB	13.8 MB

Chapter 4

Analysis of the Energy Saving in Emerging Information-Centric Metropolitan Area Networks

Starting from the protocol architecture presented in Section 3.1.1, this chapter investigates the energy saving resulting from the application of the discussed approach in Metropolitan Area Networks with respect to the reference pull-based approach.

4.1 Modeling the energy consumption

The first step towards the analysis of the energy consumption of an NDN-based network is delineating the energy consumption related to specific network actions and behaviors. In order to accomplish this, the work in [107] empirically developed an energy consumption model for softwarized NDN routers based on multi-purpose hardware. The considered model outlines how individual hardware components of routers consume energy under a range of loads. Specifically, the model separates the energy consumption in four different components, that are related to the CPU, memory, Network Interface Card, and chassis hardware parts.

The formulation in the following embraces the scenario described in Section 3.1 and considers NDN as an overlay over IP. Moreover, for the sake of simplicity, this thesis work considers the linearized model proposed in [107], but the obtained results remain valid in the considered scenario.

4.1.1 Formulation

Let Φ_{CPU}, Φ_{mem}, Φ_{NIC}, and $\Phi_{chassis}$ be the contributions to the energy consumption per second per router due to the processing activities of the CPU, the memory access, the forwarding operations of the Network Interface Card, and the chassis during the idle period, respectively. Then, the energy consumption per second per router can be modeled as:

$$\Phi_{router} = \Phi_{CPU} + \Phi_{mem} + \Phi_{NIC} + \Phi_{chassis}. \tag{4.1}$$

Theorem 5. *Let c_{NDN}, h_{p0}, λ_{NDN}, S_{NDN}, Γ_{mem}, and λ_{IP} be the amount of CPU cycles required to process the NDN packets, the frequency of the CPU, the number of NDN packets forwarded per second, the average size of an NDN packet, the ratio between the number of accessed bytes per bytes of information retrieved, and the number of IP packets forwarded per second. Then, according to [107], the energy consumption per second per router is equal to:*

$$\begin{aligned}\Phi_{router} &= \phi_{CPU}\frac{c_{NDN}}{h_{P0}} + \psi_{CPU} + \\ &+ \phi_{mem}\,\lambda_{NDN}\,S_{NDN}\,\Gamma_{mem} + \\ &+ \phi_{NIC}\,\lambda_{IP} + \\ &+ \phi_{chassis},\end{aligned} \tag{4.2}$$

where ϕ_{CPU}, ϕ_{mem}, ϕ_{NIC}, and $\phi_{chassis}$ are coefficients resulting from the linear regression on the behavior of the CPU, memory, Network Interface Card, and chassis, in the considered load range, respectively, and ψ_{CPU} is the intercept resulting from the linear regression on the behavior of the CPU in the considered load range.

Proof. The activities related to the CPU include the processing of the computing load. In this case, energy consumed by the processing of the computational load depends on the load (i.e., the CPU cycles per second required for processing NDN packets, c_{NDN}) and the clock speed of the processor, h_{P0}. By using linear regression, it is possible to model the energy consumption due to CPU activities as:

$$\Phi_{CPU} = \phi_{CPU}\frac{c_{NDN}}{h_{P0}} + \psi_{CPU}, \tag{4.3}$$

where ϕ_{CPU} and ψ_{CPU} are the coefficient and the intercept resulting from the linear regression on empirical measures. Specifically, ψ_{CPU} models the fixed energy consumption component of the CPU working at high frequencies and does not change with the computing load.

The memory access activities envisage the access to cache memory in CS and forwarding table memory in PIT and FIB when Interest and Data packets are processed. Then, the amount of memory accessed per second depends on the number of NDN packets forwarded per second, λ_{NDN}, the average size of NDN packets, S_{NDN}, and the ratio of the packet size to the size of the memory accessed per packet, Γ_{mem}. Considering contents generated in real-time, the probability to find a copy of the requested content in the cache of network routers is zero and the energy consumption due to memory access activities can be modeled as:

$$\Phi_{mem} = \phi_{mem} \, \lambda_{NDN} \, S_{NDN} \, \Gamma_{mem}, \tag{4.4}$$

where ϕ_{mem} is the coefficient related to the energy consumption per byte of accessed memory, resulting from the linear regression on empirical measures.

The activities related to the Network Interface Card include the forwarding of IP packets. Hence, the energy consumption depends on the amount of IP packets forwarded per second at the network layer, λ_{IP}.

$$\Phi_{NIC} = \phi_{NIC} \lambda_{IP}, \tag{4.5}$$

where ϕ_{NIC} is the coefficient related to the energy consumption per forwarded packet, resulting from the linear regression on empirical measures.

Finally, the energy consumed by the chassis during the idle period depends only on the hardware configuration. Hence,it holds that:

$$\Phi_{chassis} = \phi_{chassis}, \tag{4.6}$$

where $\phi_{chassis}$ is the energy consumed per second by the router when the CPU, the memory, and the Network Interface Card are idle.

Now, by substituting (4.3), (4.4), (4.5), and (4.6) in (4.1), it is possible to prove the theorem.

$\square$

A summary of the main symbols adopted is presented in Table 4.1.

TABLE 4.1: List of adopted symbols.

Symbol	**Description**
Φ_{router}	Energy consumption per second per router
Φ_{CPU}	Energy consumption per second per router due to the processing activities of the CPU
Φ_{mem}	Energy consumption per second per router due to the memory access
Φ_{NIC}	Energy consumption per second per router due to the forwarding operations of the Network Interface card
$\Phi_{chassis}$	Energy consumption per second per router due to the chassis during the idle period
ϕ_{CPU}	Coefficient resulting from the linear regression on the behavior of the CPU
ϕ_{mem}	Coefficient resulting from the linear regression on the behavior of the memory
ϕ_{NIC}	Coefficient resulting from the linear regression on the behavior of the Network Interface Card
$\phi_{chassis}$	Coefficient resulting from the linear regression on the behavior of the chassis
c_{NDN}	CPU cycles required to process the NDN packets
h_{p0}	Clock frequency of the CPU
ψ_{CPU}	Intercept resulting from the linear regression on the behavior of the chassis
λ_{NDN}	Number of NDN packets forwarded per second by the router
λ_{IP}	Number of IP packets forwarded per second by the router
S_{NDN}	Average size of NDN packets
Γ_{mem}	Ratio of the packet size to the size of the memory accessed per packet

4.1.2 Main parameters settings

The numerical analysis considers a MATLAB script to simulate mobility, implements the data exchanges described in Section 3.1.1, computes the energy consumption by using the model presented in [107], and quantifies the daily energy saving achieved with the investigated protocol.

The average amount of daily energy saving of each component is calculated as the difference between the energy consumption related to reference NDN pull-based approaches and the energy consumption related to the proposed protocol. Finally, the average amount of overall daily energy saving ΔE_d is calculated as the sum of the average amount of daily energy saving related to the CPU (ΔE_{CPU}), memory (ΔE_{mem}), and Network Interface Card (ΔE_{NIC}) components. Fixed consumption due to the idle consumption of network devices (i.e., the chassis component) are neglected in the analysis as they are not affected by the investigated protocol.

As in [107], the maximum size of IP packets is set to 1500 B. Considering Interest and Data packets' header sizes are set according to NDN specifications in [2], each Interest and control related NDN message can be sent as the payload of a single IP packet, while bigger Data packets must be fragmented over multiple IP packets.

Three different hardware configurations are considered for the softwarized NDN routers. The hardware specifications are summarized in Table 4.2.

TABLE 4.2: Summary of hardware configurations considered in [107].

	Configuration 1	Configuration 2	Configuration 3
CPU	2 Intel Xeon E5620 (2.4 GHz x 4 cores)	Intel Xeon E3-1220 (3.10 GHz x 4 cores)	Intel Itanium 9520 (1.73 GHz x 4 cores)
Memory	12 GB DDR3	16 GB DDR3	8 GB DDR3
Network Interface Card	Intel X540-T2 10GBASE-T	Intel X540-T2 10GBASE-T	HP Integrity I/O Ethernet adapter

A list of the main parameters used to implement the model and compute the energy consumption for each hardware configuration is presented in Table 4.3.

The analysis envisaged the average time interval between the generation of consecutive contents, namely T_D, ranging from 0.1 s to 10000 s and content sizes equal to 3 MB, 30 MB, 300 MB and 3 GB. These values are chosen to cope with bursty applications, which could heavily harm network performance. The conducted simulations also considered a 10 km x 10 km urban

TABLE 4.3: List of parameters for hardware configurations [107].

	Configuration 1	Configuration 2	Configuration 3
ϕ_{CPU} [W/cycle]	5.88	5.85	1.65
ψ_{CPU} [W]	23.58	5.43	8.15
ϕ_{mem} [J/B]	$0.61 * 10^-9$	$0.61 * 10^-9$	$0.61 * 10^-9$
Γ_{mem}	35.67	35.67	35.67
ϕ_{NIC} [J/packet]	$1.26 * 10^-5$	$1.26 * 10^-5$	$1.26 * 10^-5$
$\phi_{chassis}$ [W]	34.20	68.90	117.82

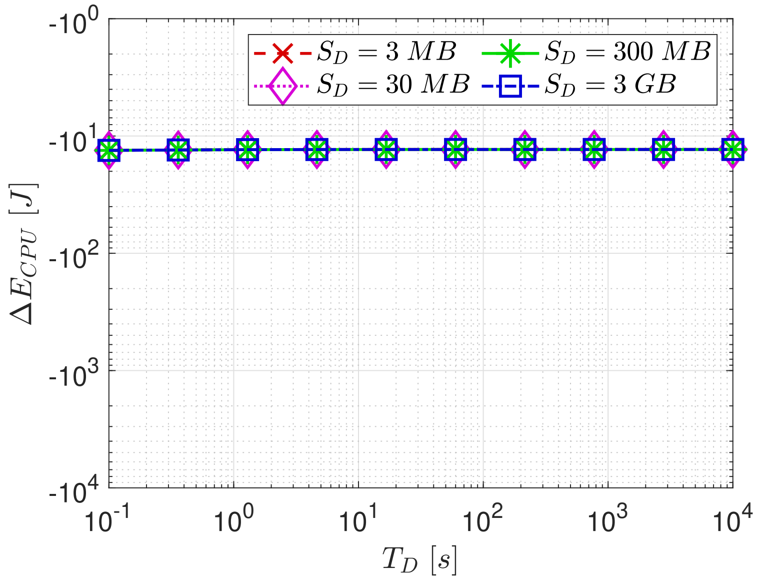
area, two different average cell radius of network attachment points r equal to 50m and 150m, and two different consumer speed values, being 3 km/h and 30 km/h. Network attachment points are connected to a core network composed by N nodes, where N is equal to 1415 and 12732 for $r = 150$ m and $r = 50$ m, respectively. Those nodes are connected according to scale-free networks theory, where nodes are distributed with a power law factor $\gamma = 3$ and each new node is connected to two previously existing nodes, following the preferential attachment law [87]. Several topologies were generated with these characteristics with the BRITE. Then, random positions are considered for the consumer, producer, and SDC and mobility is simulated following a random waypoint model with a MATLAB script. The outcome of the conducted simulations is averaged on 30000 realizations.

4.2 Analysis of the energy consumption components

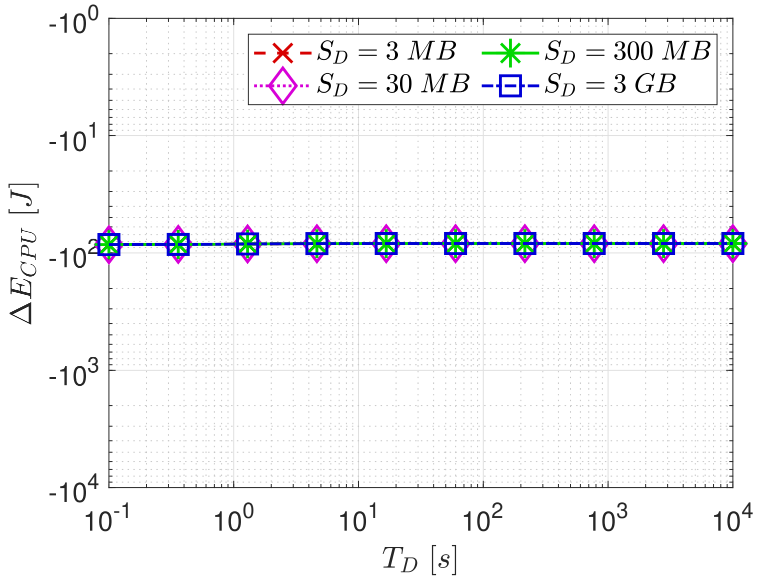
Figures from 4.1 to 4.9 show the average amount of daily energy saving per consumer due to the investigated protocol for the three hardware configurations, reporting the contribution of each hardware component separately.

Specifically, Figures 4.1, 4.2, and 4.3 illustrate the average amount of daily energy saving related to the CPU component due to the investigated protocol for different hardware configurations, network sizes, consumer speed values, and application settings.

The analysis highlights that the investigated protocol increases the consumption of the CPU hardware elements of a negligible amount in all the considered scenarios, as softwarized routers have to process an higher number of packets introduced by the investigated protocol. In fact, the energy saving increases as the consumer speed, v, decrease and the cell radius, r,



(A) $r = 150\ m$, $v = 3\ km/h$



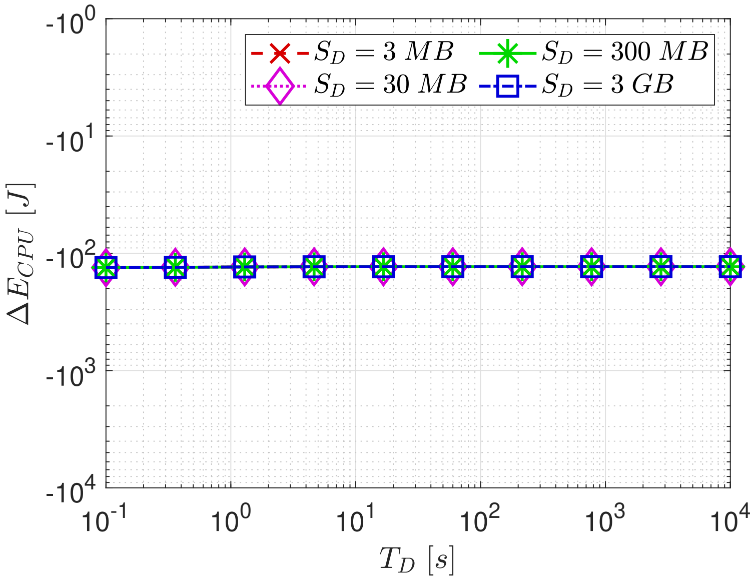
(B) $r = 50\ m$, $v = 3\ km/h$

FIGURE 4.1: Average amount of daily energy saving related to the CPU component for Configuration 1 due to the investigated protocol, as a function of network size, consumer speed, and application settings.

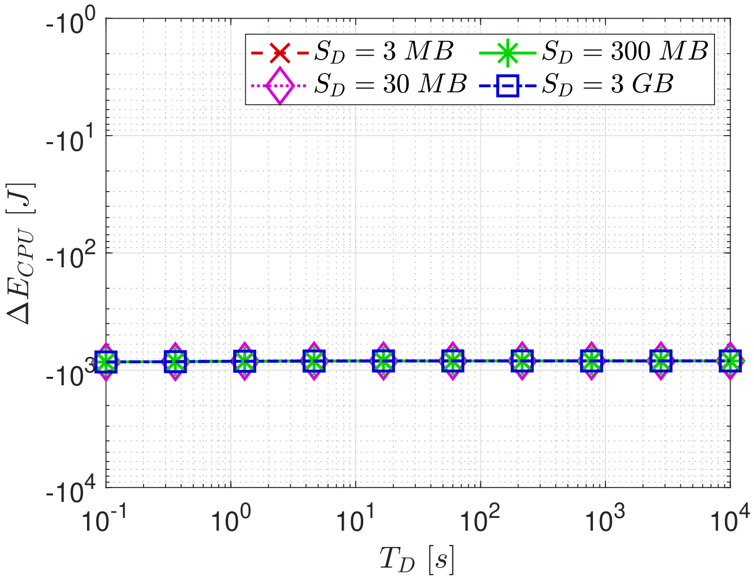
increases, due to the lower number of times the handover procedures are triggered.

The investigation also reveals that the energy saving due to the CPU is stable with respect to the content size, S_D, and the average time between the generation of consecutive contents, T_D.

The main reason behind this behavior is the high energy consumption component of a CPU working at high frequency (i.e., ψ_{CPU}), regardless of the



(C) $r = 150\ m, v = 30\ km/h$

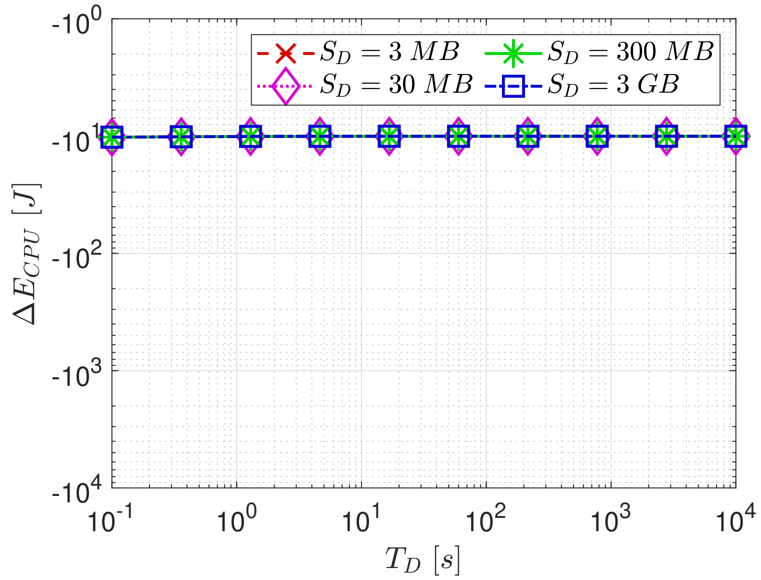


(D) $r = 50\ m, v = 30\ km/h$

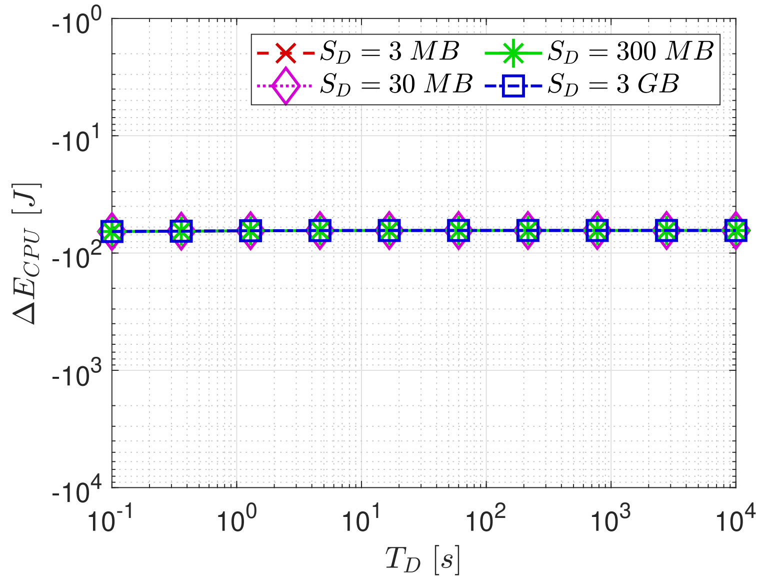
FIGURE 4.1: Average amount of daily energy saving related to the CPU component for Configuration 1 due to the investigated protocol, as a function of network size, consumer speed, and application settings.

computing load. In this case, the energy consumption due to the processing of NDN packets is negligible with respect to the fixed component of CPU consumption.

Finally, hardware configuration 1 presents lower energy savings with respect to configurations 2 and 3. This is due to the higher consumption of the considered CPU hardware (see Table 4.2).

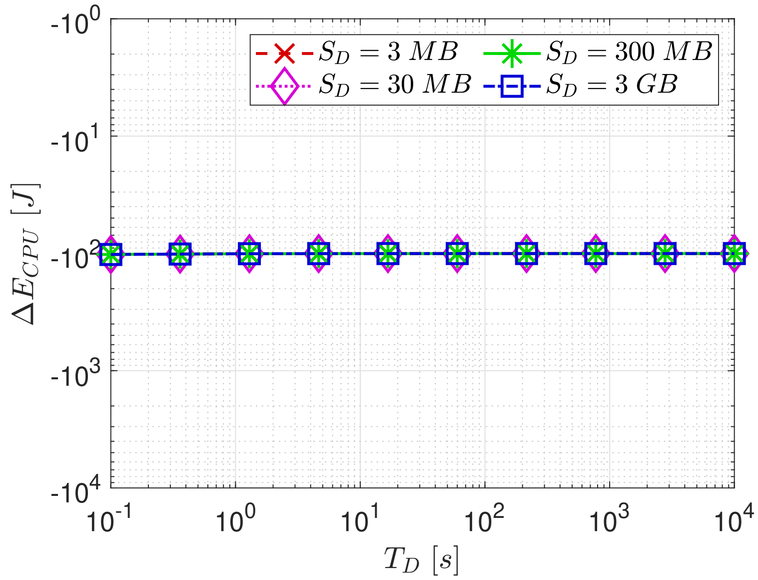


(A) $r = 150\ m, v = 3\ km/h$

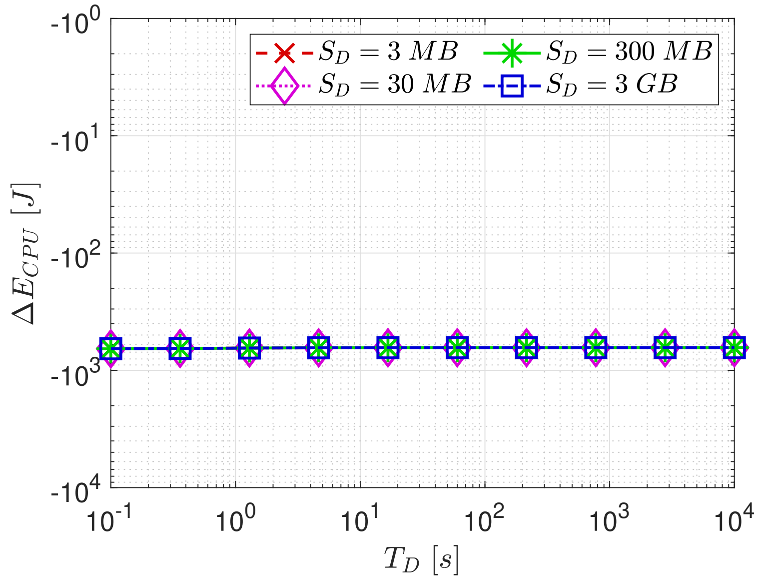


(B) $r = 50\ m, v = 3\ km/h$

FIGURE 4.2: Average amount of daily energy saving related to the CPU component for Configuration 2 due to the investigated protocol, as a function of network size, consumer speed, and application settings.

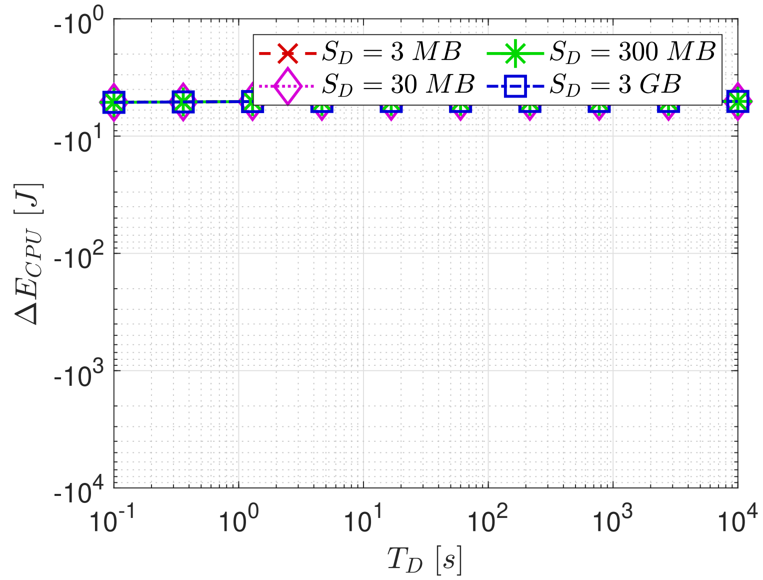


(C) $r = 150\ m, v = 30\ km/h$

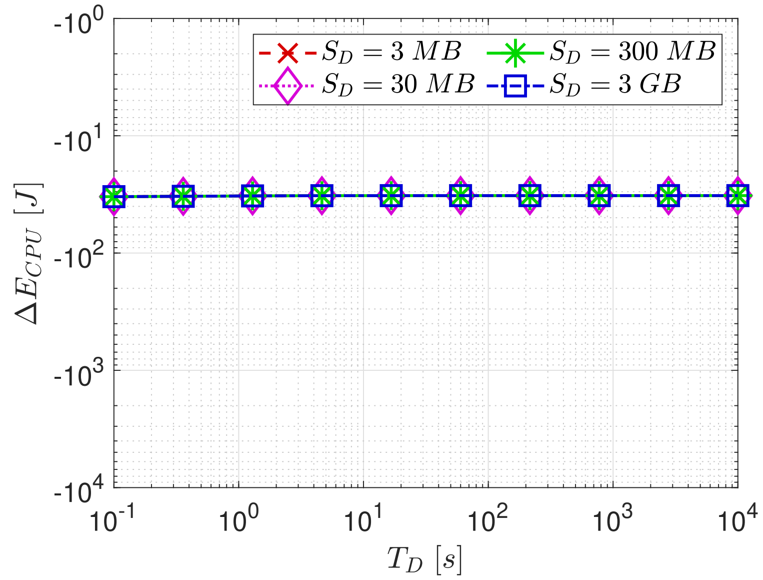


(D) $r = 50\ m, v = 30\ km/h$

FIGURE 4.2: Average amount of daily energy saving related to the CPU component for Configuration 2 due to the investigated protocol, as a function of network size, consumer speed, and application settings.

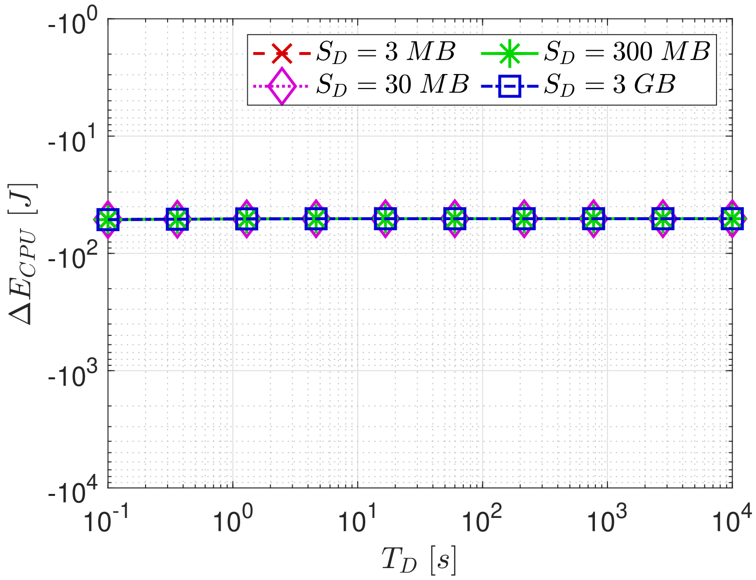


(A) $r = 150\ m, v = 3\ km/h$

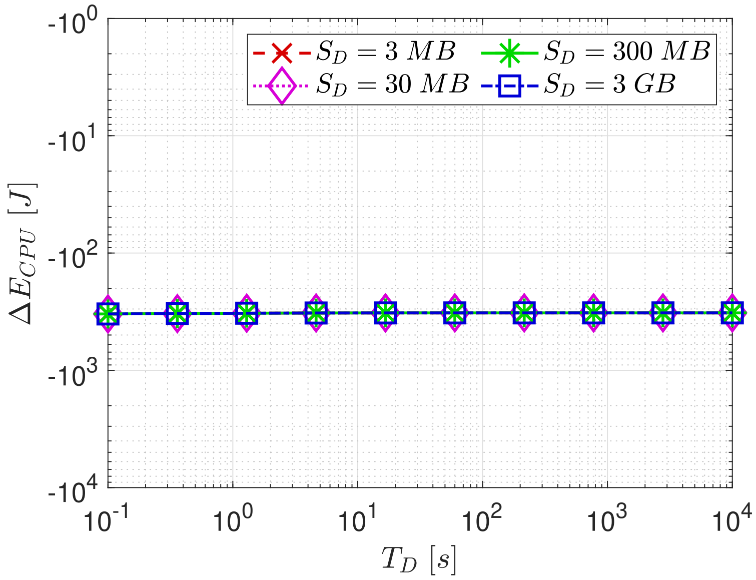


(B) $r = 50\ m, v = 3\ km/h$

FIGURE 4.3: Average amount of daily energy saving related to the CPU component for Configuration 3 due to the investigated protocol, as a function of network size, consumer speed, and application settings.



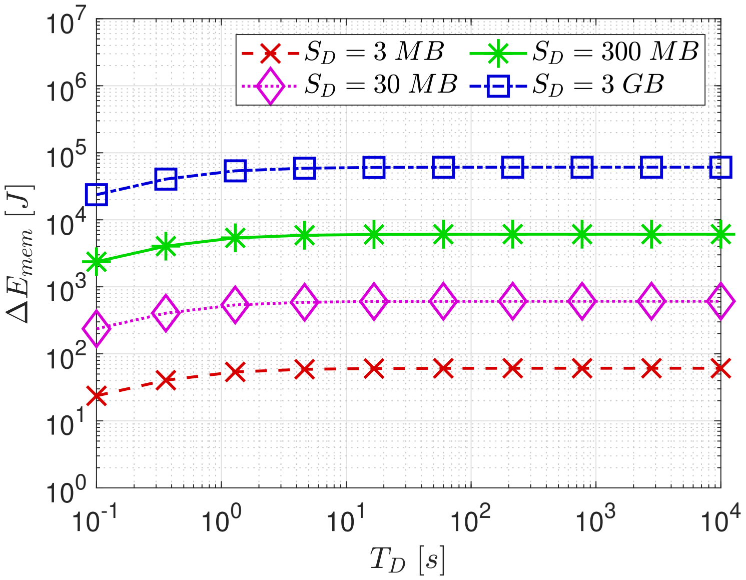
(C) $r = 150\ m, v = 30\ km/h$



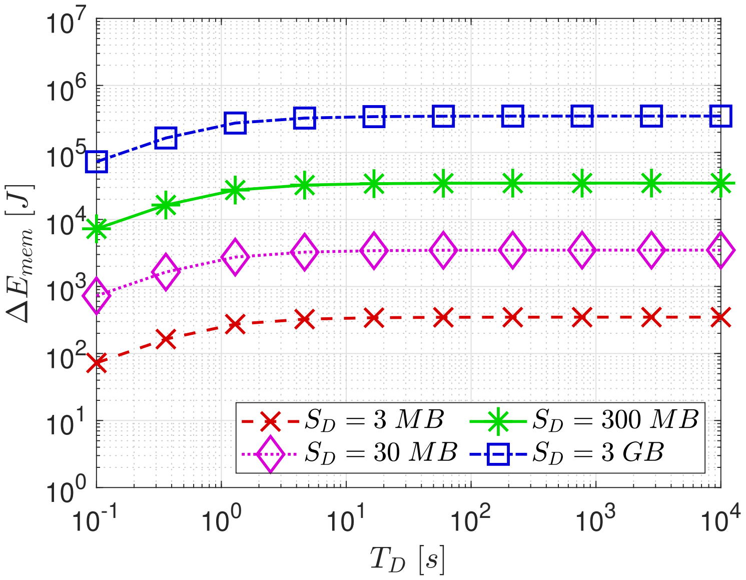
(D) $r = 50\ m, v = 30\ km/h$

FIGURE 4.3: Average amount of daily energy saving related to the CPU component for Configuration 3 due to the investigated protocol, as a function of network size, consumer speed, and application settings.

Figures 4.4, 4.5, and 4.6 illustrate the average amount of daily energy saving related to the memory component due to the investigated protocol for different hardware configurations, network sizes, consumer speed values, and application settings.



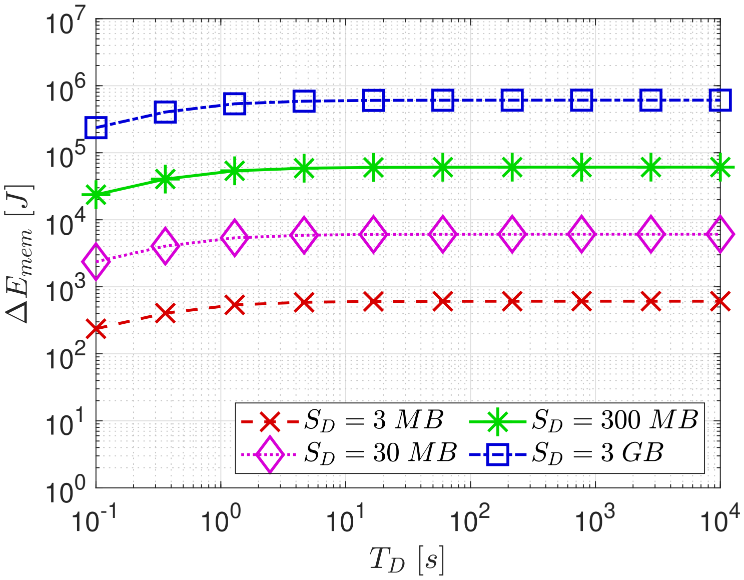
(A) $r = 150\ m$, $v = 3\ km/h$



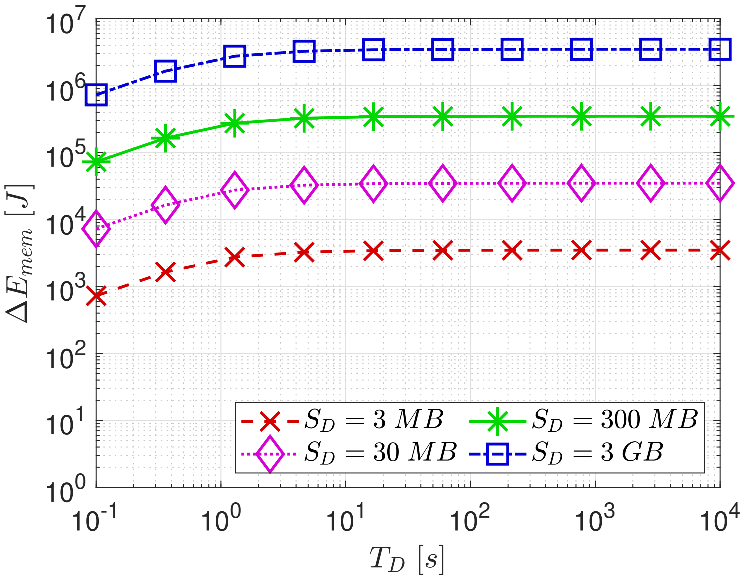
(B) $r = 50\ m$, $v = 3\ km/h$

FIGURE 4.4: Average amount of daily energy saving related to the memory component for Configuration 1 due to the investigated protocol, as a function of network size, consumer speed, and application settings.

The analysis highlights that the investigated protocol vastly increases the energy savings in all the considered scenarios.



(C) $r = 150$ m, $v = 30$ km/h

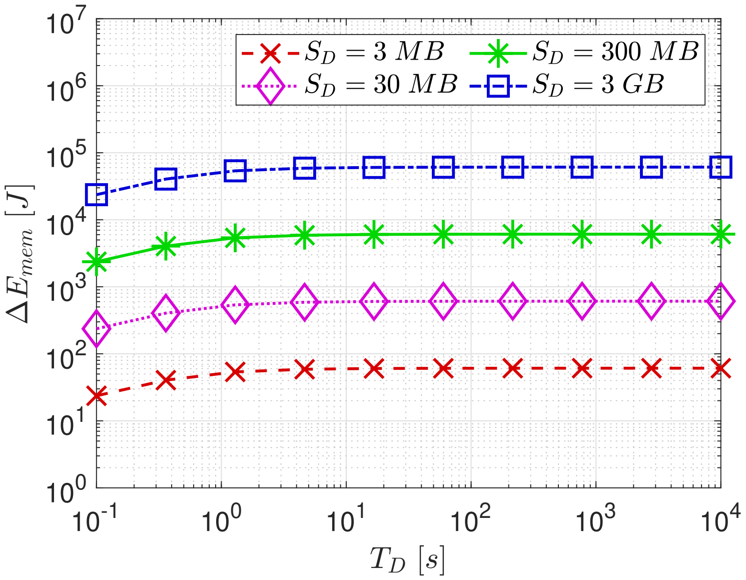


(D) $r = 50$ m, $v = 30$ km/h

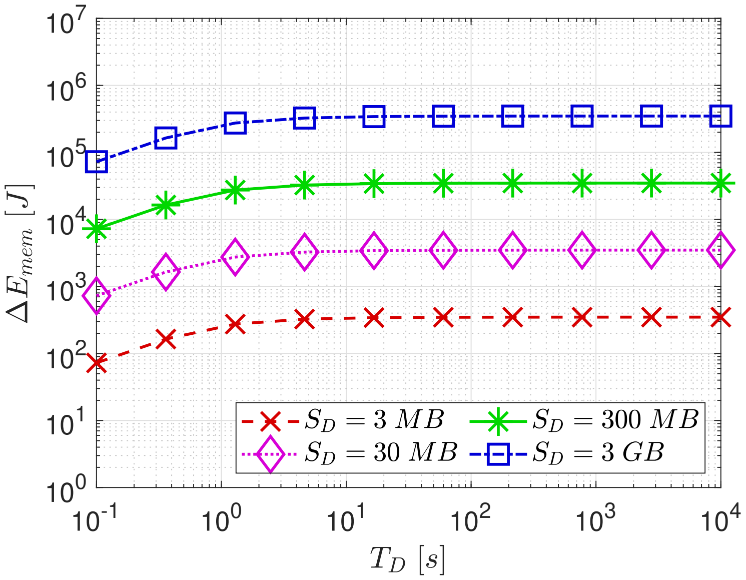
FIGURE 4.4: Average amount of daily energy saving related to the memory component for Configuration 1 due to the investigated protocol, as a function of network size, consumer speed, and application settings.

Specifically, the energy saving increases as the consumer speed increases and the cell radius decreases, due to the higher number of times the handover procedures are triggered.

The energy saving due to the memory also increases with the content size and the average time between the generation of consecutive contents, as the SDC is able to prevent the unnecessary caching of an higher amount of contents data.



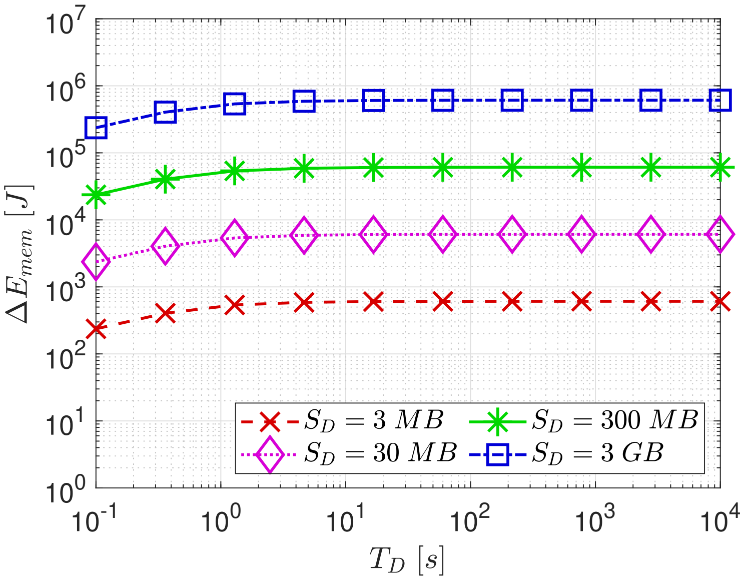
(A) $r = 150\ m, v = 3\ km/h$



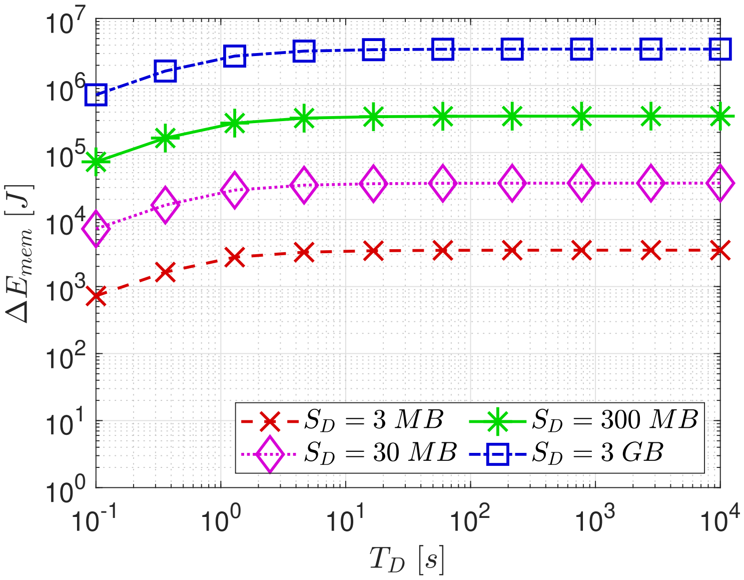
(B) $r = 50\ m, v = 3\ km/h$

FIGURE 4.5: Average amount of daily energy saving related to the memory component for Configuration 2 due to the investigated protocol, as a function of network size, consumer speed, and application settings.

Moreover, the energy saving remains unchanged among configurations due to the same type of hardware component used (see Table 4.2).

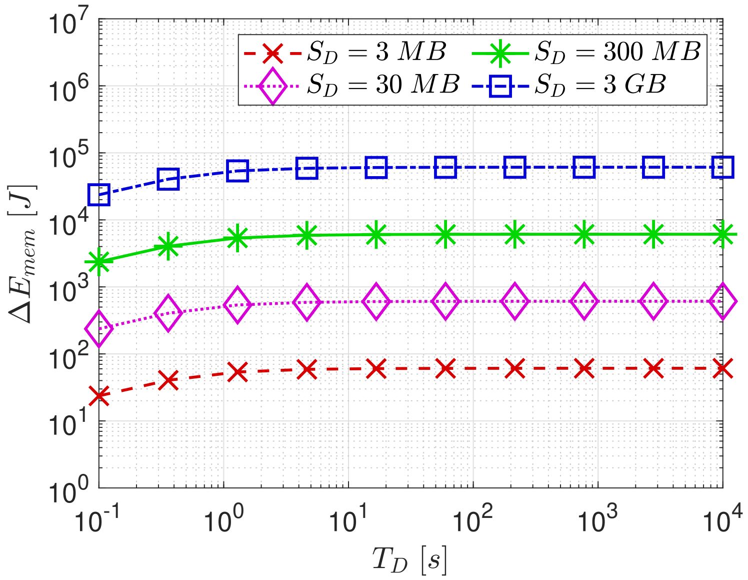


(C) $r = 150\ m$, $v = 30\ km/h$

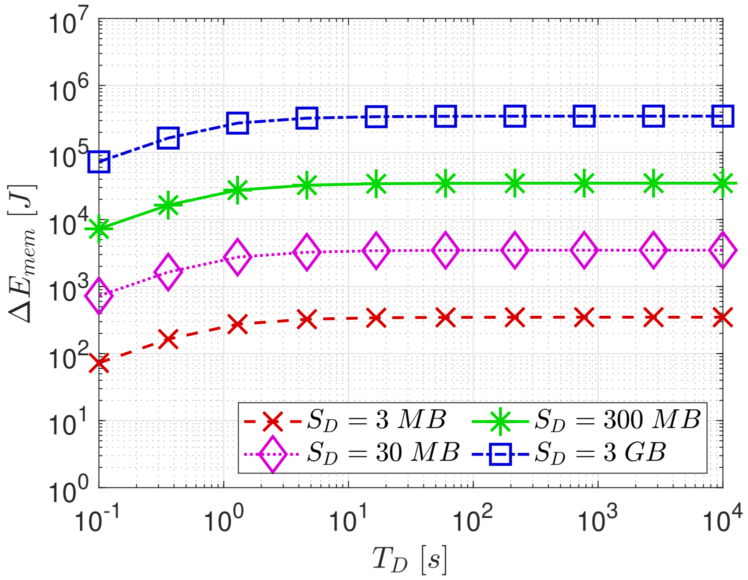


(D) $r = 50\ m$, $v = 30\ km/h$

FIGURE 4.5: Average amount of daily energy saving related to the memory component for Configuration 2 due to the investigated protocol, as a function of network size, consumer speed, and application settings.

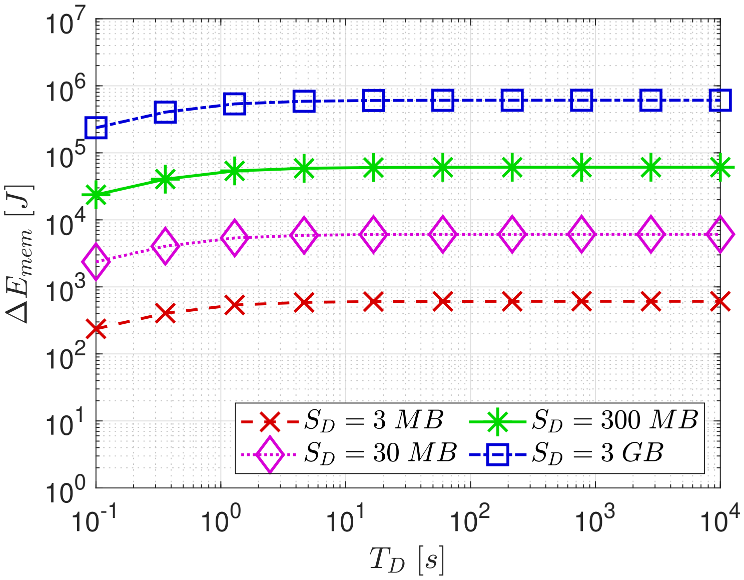


(A) $r = 150\ m, v = 3\ km/h$

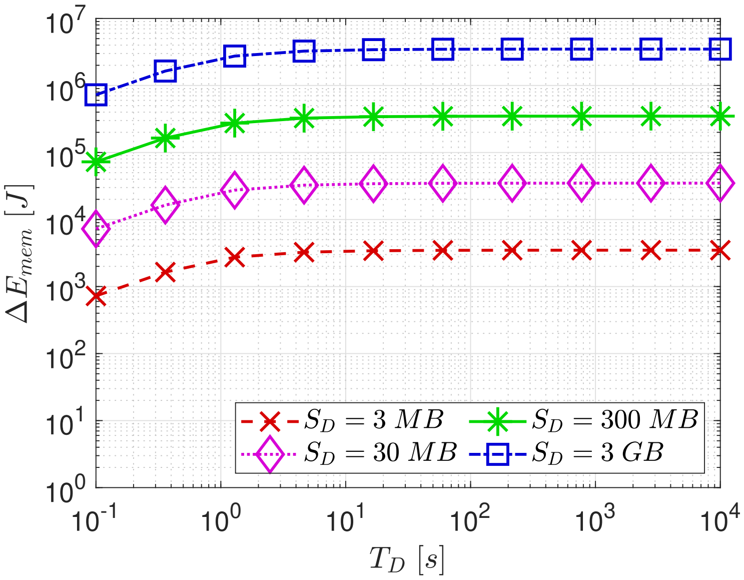


(B) $r = 50\ m, v = 3\ km/h$

FIGURE 4.6: Average amount of daily energy saving related to the memory component for Configuration 3 due to the investigated protocol, as a function of network size, consumer speed, and application settings.



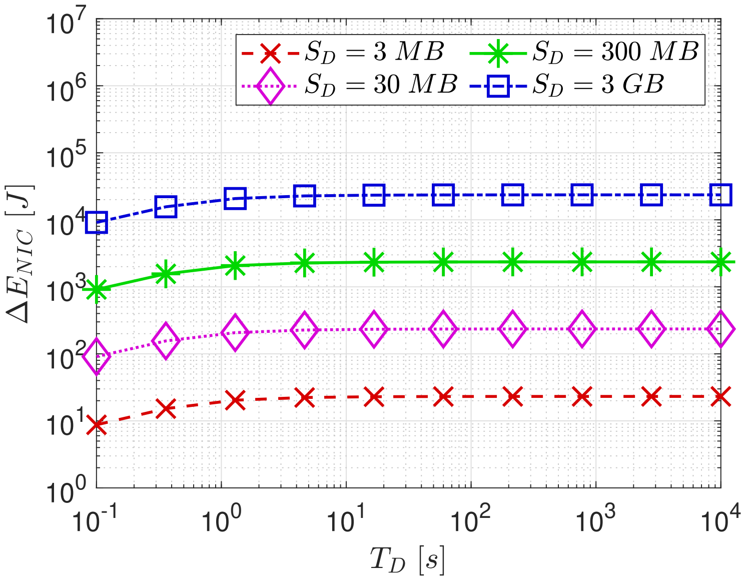
(C) $r = 150\ m, v = 30\ km/h$



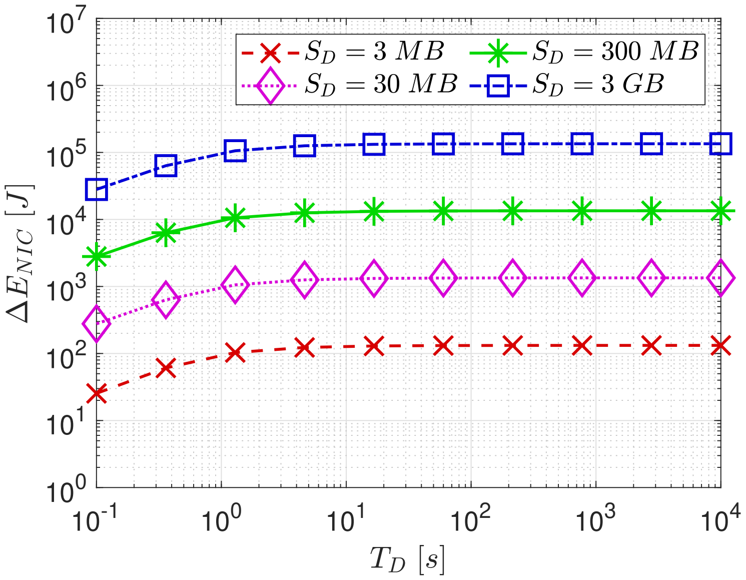
(D) $r = 50\ m, v = 30\ km/h$

FIGURE 4.6: Average amount of daily energy saving related to the memory component for Configuration 3 due to the investigated protocol, as a function of network size, consumer speed, and application settings.

Figures 4.7, 4.8, and 4.9 illustrate the average amount of daily energy saving related to the Network Interface Card component due to the investigated protocol for different hardware configurations, network sizes, consumer speed values, and application settings.



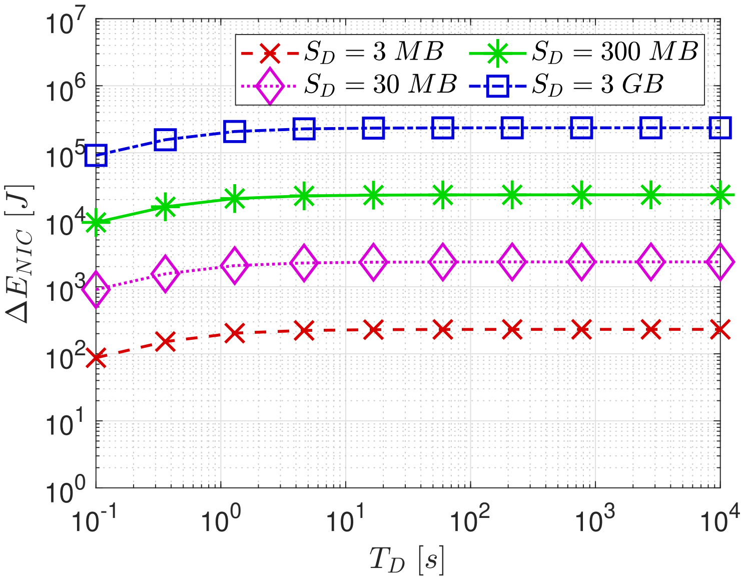
(A) $r = 150\ m, v = 3\ km/h$



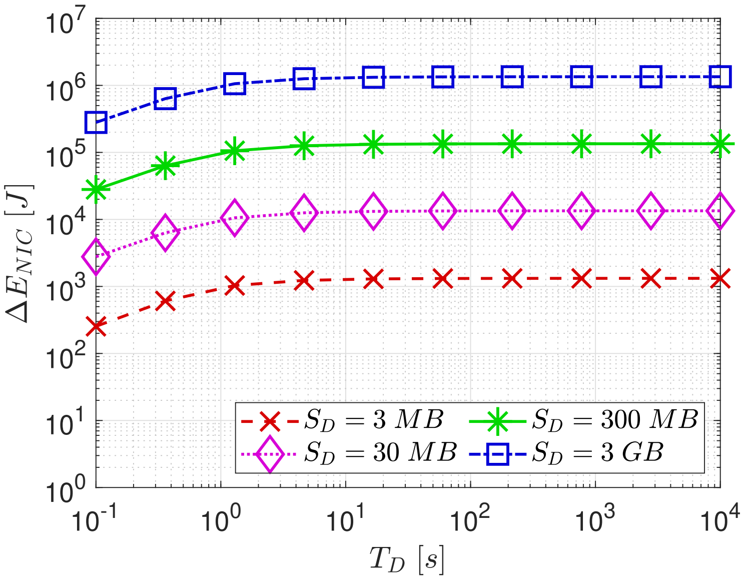
(B) $r = 50\ m, v = 3\ km/h$

FIGURE 4.7: Average amount of daily energy saving related to the Network Interface Card component for Configuration 1 due to the investigated protocol, as a function of network size, consumer speed, and application settings.

Again, the analysis highlights that the investigated protocol vastly increases the energy savings in all the considered scenarios.



(C) $r = 150\ m$, $v = 30\ km/h$

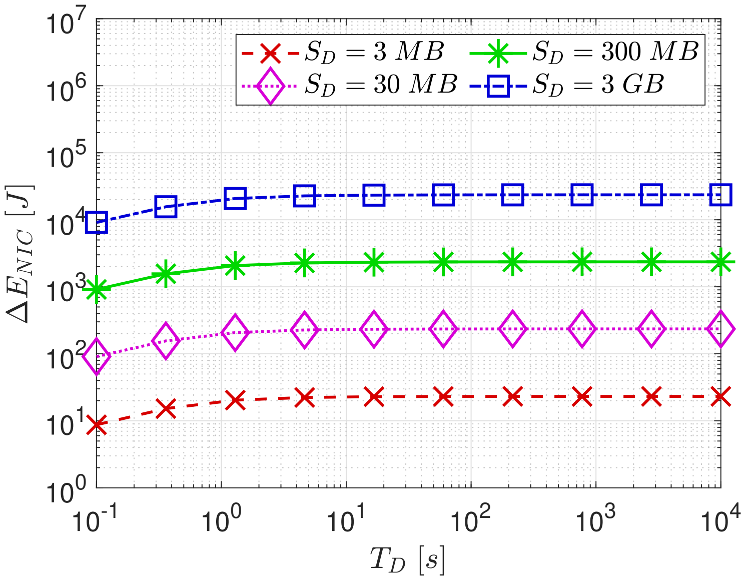


(D) $r = 50\ m$, $v = 30\ km/h$

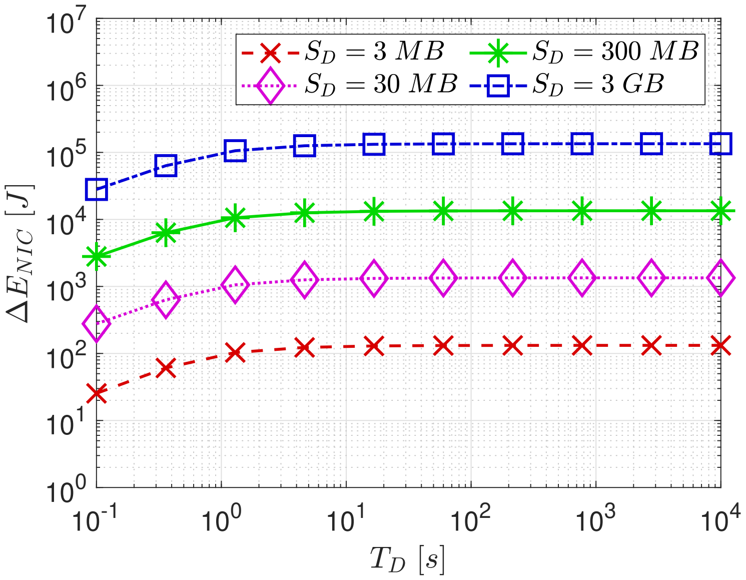
FIGURE 4.7: Average amount of daily energy saving related to the Network Interface Card component for Configuration 1 due to the investigated protocol, as a function of network size, consumer speed, and application settings.

Specifically, the energy saving increases as the consumer speed increases and the cell radius decreases, due to the higher number of times the handover procedures are triggered.

The energy saving due to the Network Interface Card also increases with the content size and the average time between the generation of consecutive contents, as the SDC is able to prevent the unnecessary forwarding of an higher amount of packets. It is also worth to note that the number of packet



(A) $r = 150\ m, v = 3\ km/h$

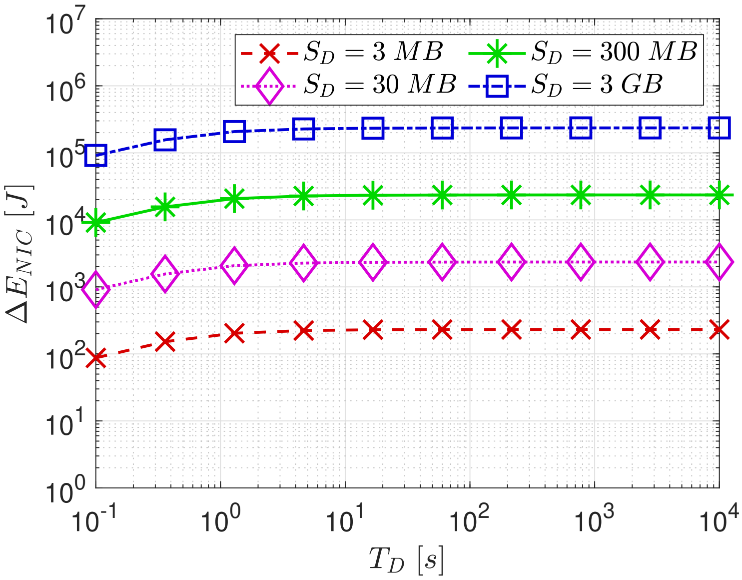


(B) $r = 50\ m, v = 3\ km/h$

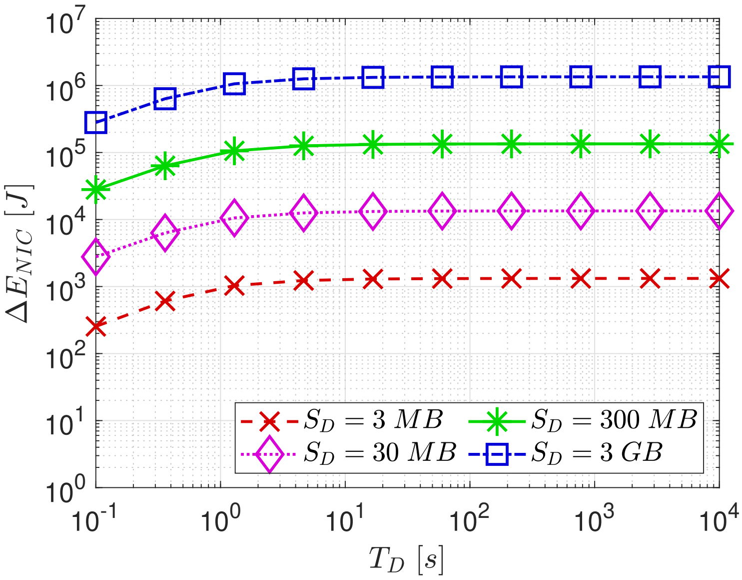
FIGURE 4.8: Average amount of daily energy saving related to the Network Interface Card component for Configuration 2 due to the investigated protocol, as a function of network size, consumer speed, and application settings.

forwarded in the network increases with the content size, as the IP payload limit, set to $1500B$, induces fragmentation.

Finally, the energy saving remains unchanged among configurations due to the same energy consumption per forwarded packet.

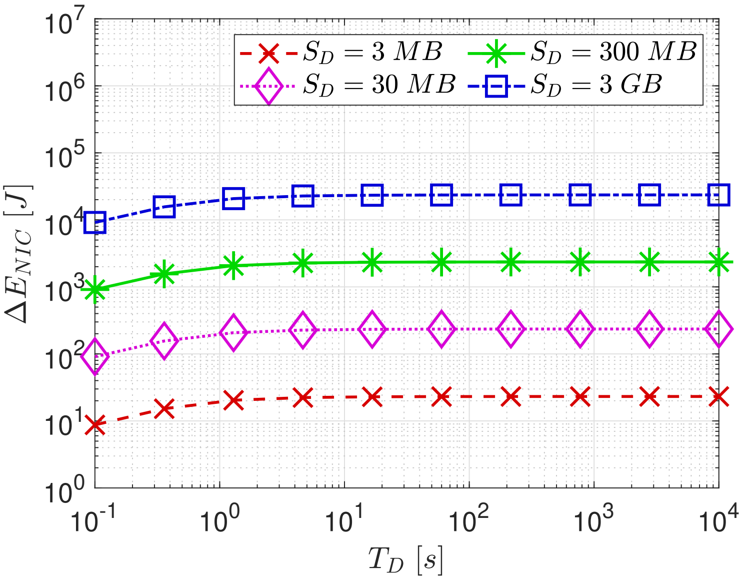


(C) $r = 150\ m, v = 30\ km/h$

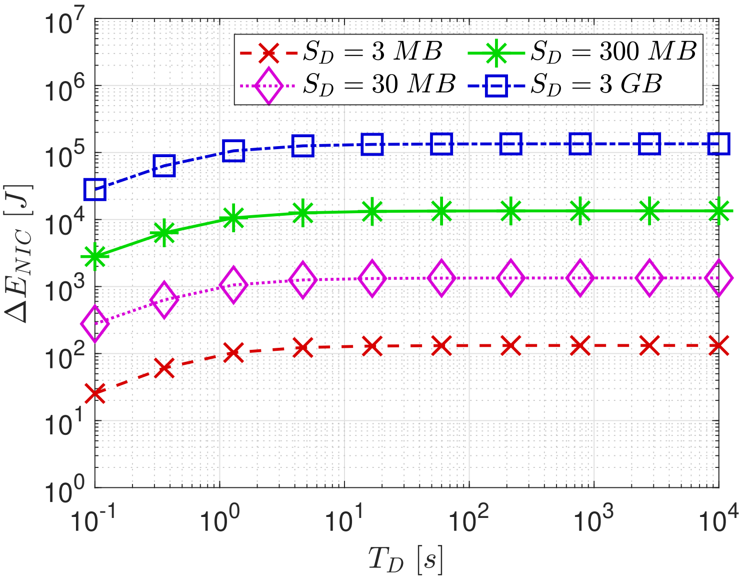


(D) $r = 50\ m, v = 30\ km/h$

FIGURE 4.8: Average amount of daily energy saving related to the Network Interface Card component for Configuration 2 due to the investigated protocol, as a function of network size, consumer speed, and application settings.

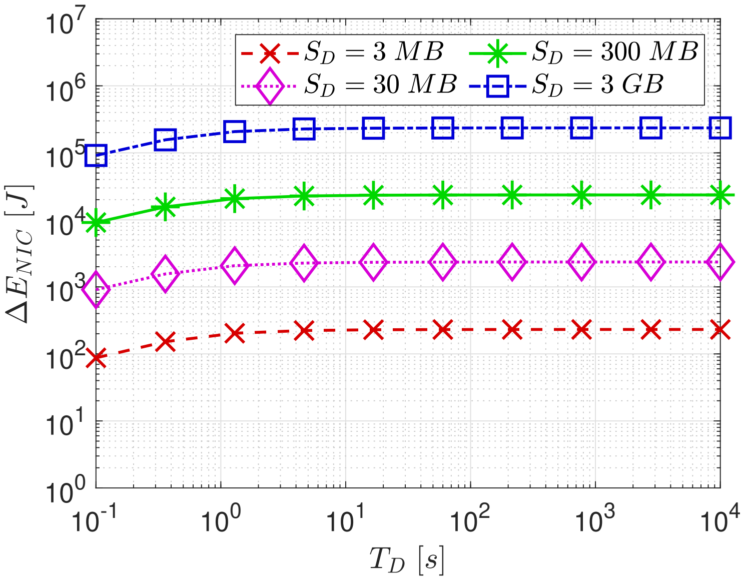


(A) $r = 150\ m, v = 3\ km/h$

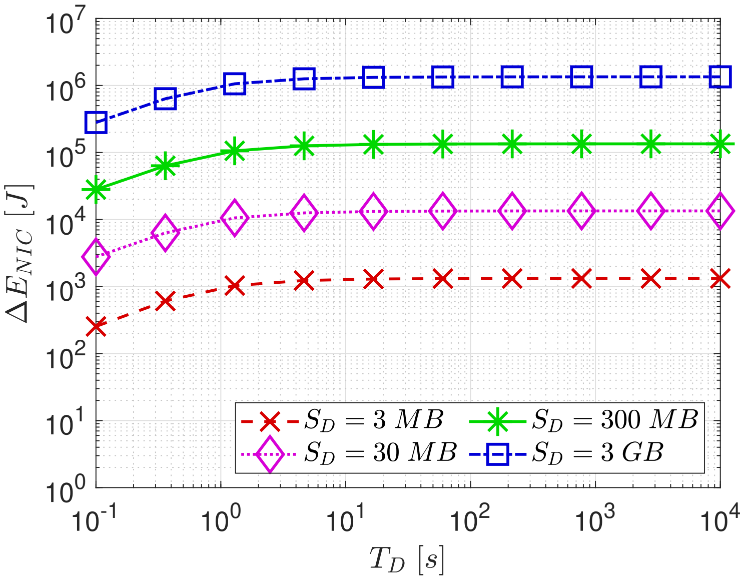


(B) $r = 50\ m, v = 3\ km/h$

FIGURE 4.9: Average amount of daily energy saving related to the Network Interface Card component for Configuration 3 due to the investigated protocol, as a function of network size, consumer speed, and application settings.



(C) $r = 150\ m, v = 30\ km/h$



(D) $r = 50\ m, v = 30\ km/h$

FIGURE 4.9: Average amount of daily energy saving related to the Network Interface Card component for Configuration 3 due to the investigated protocol, as a function of network size, consumer speed, and application settings.

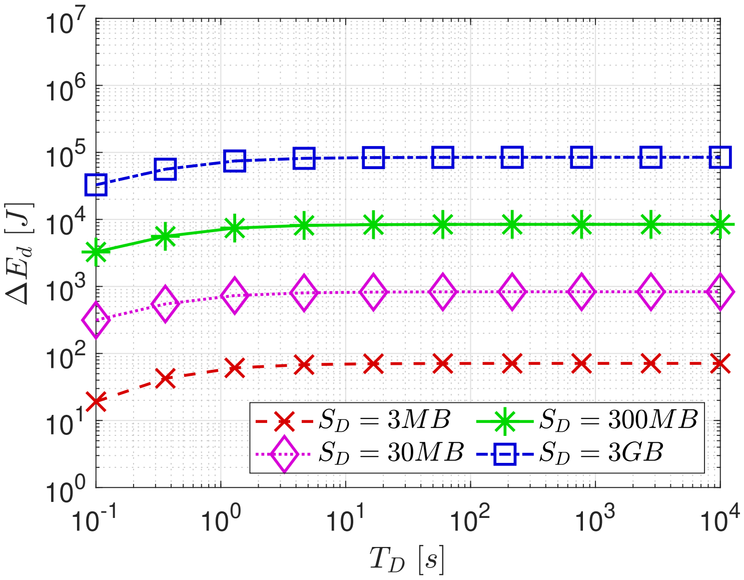
4.3 Analysis of the overall energy saving

The contribution related to the memory access has paramount importance in saving energy, as the memory component makes up for the most part of the energy savings. This is due to the proposed protocol preventing the forwarding of unnecessary packets, reducing the amount of lookups to PIT and FIB tables and the caching of the contents carried by unuseful data packets.

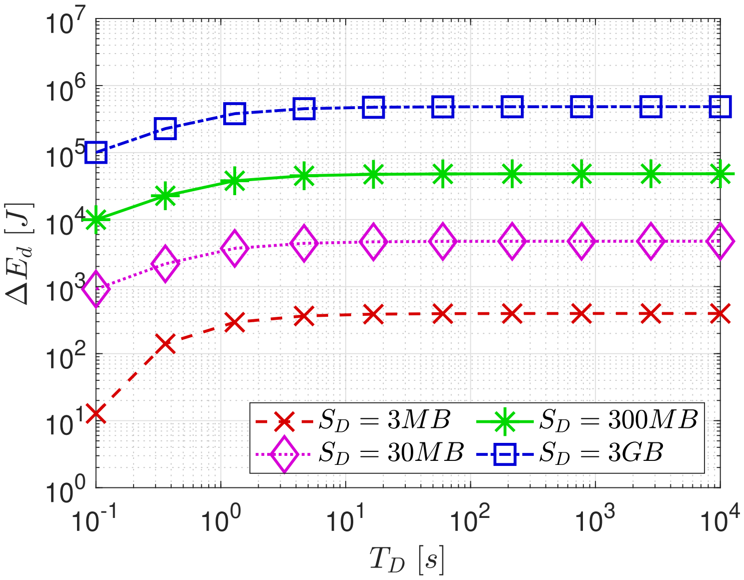
Figures 4.10, 4.11, and 4.12 show the average amount of overall daily energy saving per consumer due to the investigated protocol for different hardware configurations, network sizes, consumer speed values, and application settings. Fittingly, the proposed protocol is able to save energy in all the considered scenarios.

It is clear how the energy savings increases with the average consumer speed and decreases as the average cell radius increases. This is due to the mobile consumer changing cell more often and causing handover-related message exchanges to occur more frequently. Concurrently, the amount of overall daily energy saving increases with the average time between the generation of consecutive contents and the content size.

Finally, the differences in the energy saving between hardware configurations are negligible. This behavior is related to the different impacts on the overall daily energy saving of the hardware components. More specifically, the memory component represents the primary contribution on the energy savings and its contribution is the same for all the considered hardware configurations.

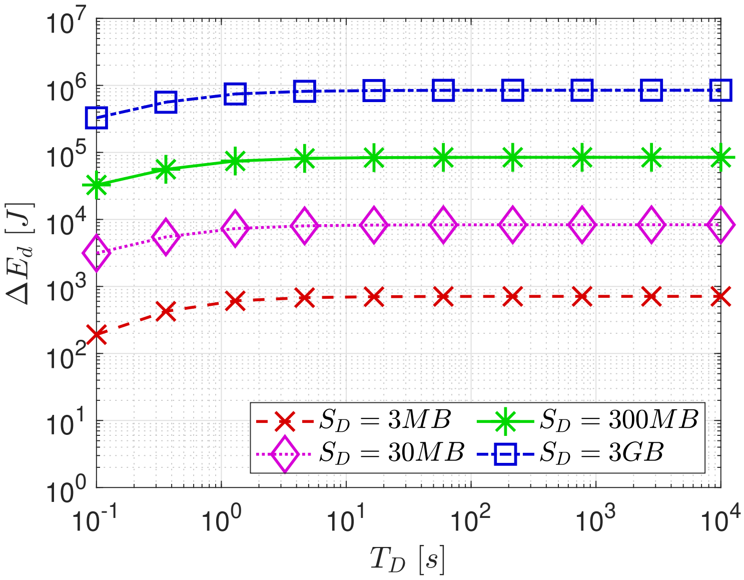


(A) $r = 150\ m, v = 3\ km/h$

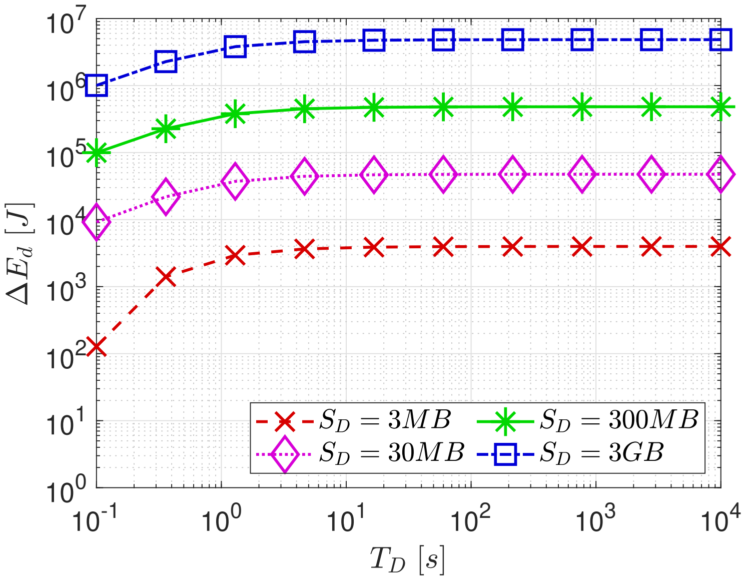


(B) $r = 50\ m, v = 3\ km/h$

FIGURE 4.10: Average amount of overall daily energy saving for Configuration 1, as a function of network size, consumer speed, and application settings.

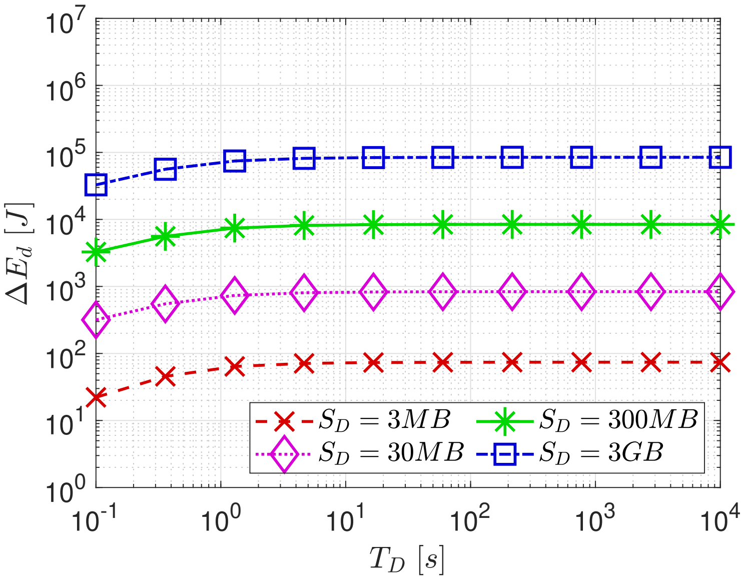


(C) $r = 150\ m, v = 30\ km/h$

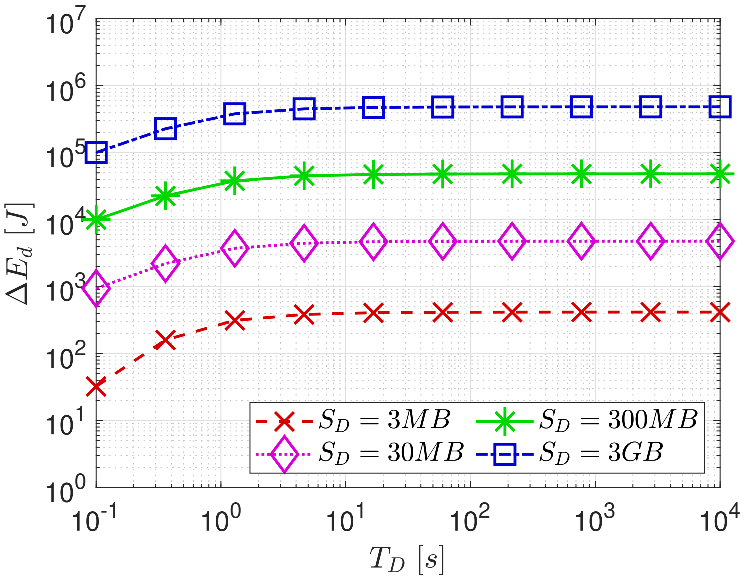


(D) $r = 50\ m, v = 30\ km/h$

FIGURE 4.10: Average amount of overall daily energy saving for Configuration 1, as a function of network size, consumer speed, and application settings.

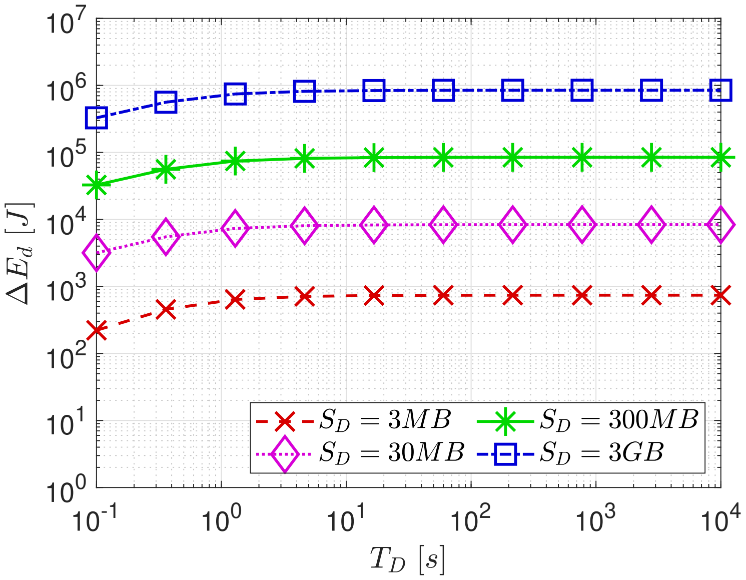


(A) $r = 150\ m, v = 3\ km/h$

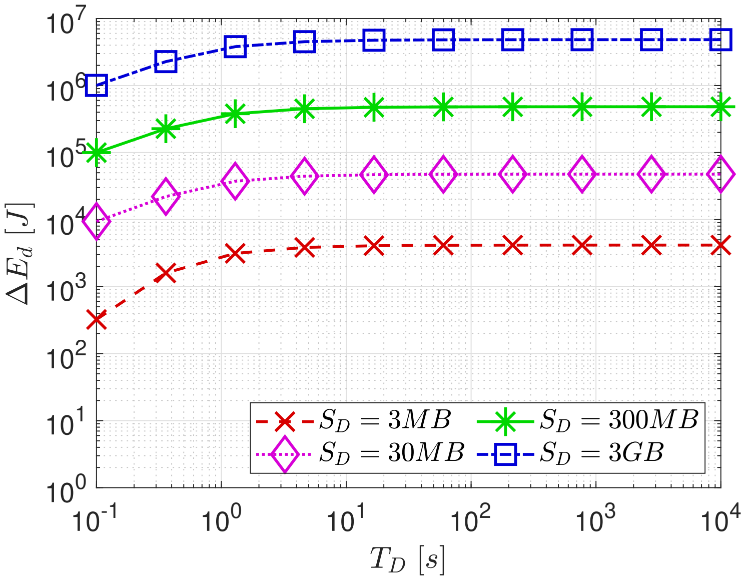


(B) $r = 50\ m, v = 3\ km/h$

FIGURE 4.11: Average amount of overall daily energy saving for Configuration 2, as a function of network size, consumer speed, and application settings.

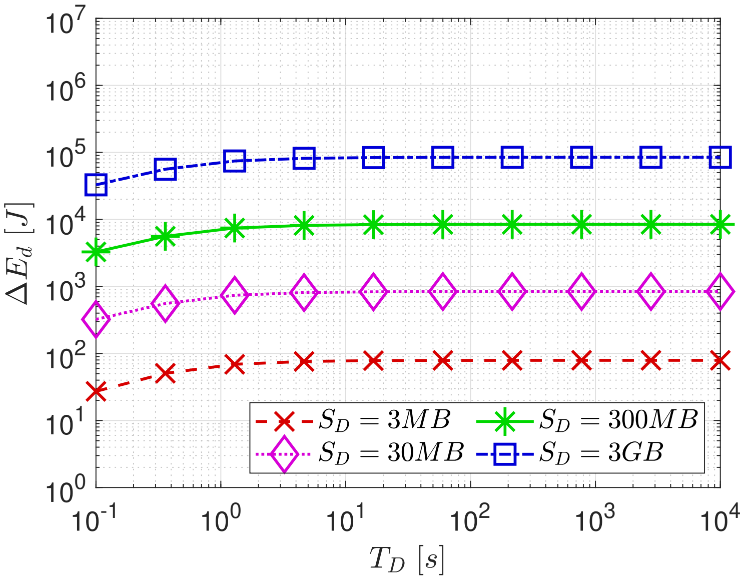


(C) $r = 150\ m, v = 30\ km/h$

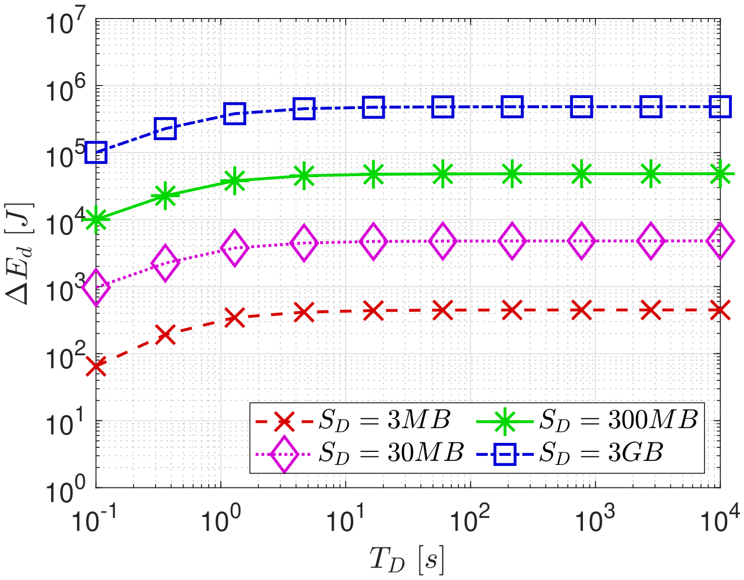


(D) $r = 50\ m, v = 30\ km/h$

FIGURE 4.11: Average amount of overall daily energy saving for Configuration 2, as a function of network size, consumer speed, and application settings.

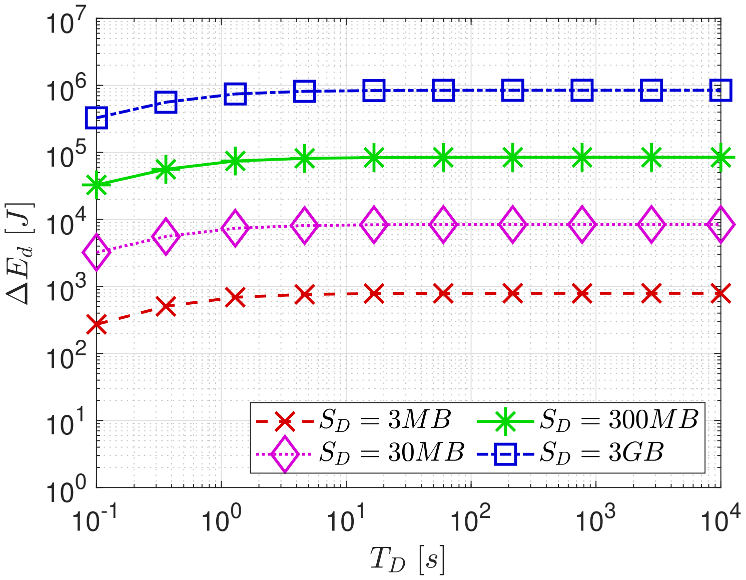


(A) $r = 150\ m, v = 3\ km/h$

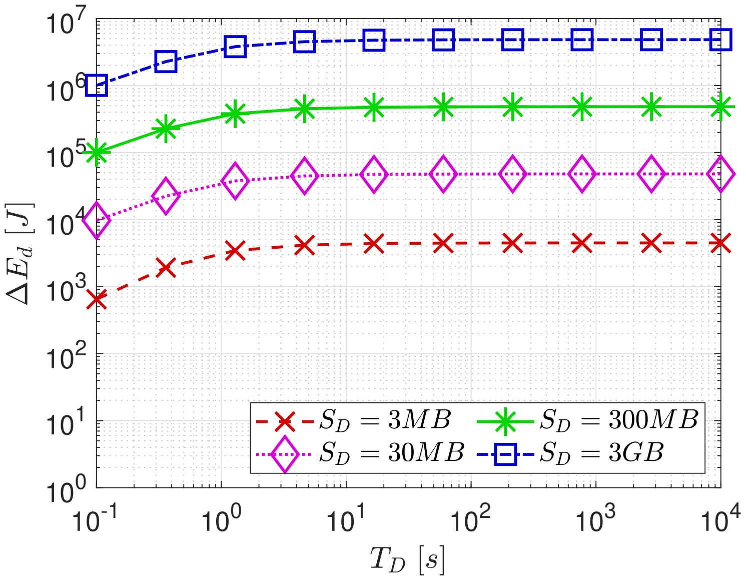


(B) $r = 50\ m, v = 3\ km/h$

FIGURE 4.12: Average amount of overall daily energy saving for Configuration 3, as a function of network size, consumer speed, and application settings.



(C) $r = 150\ m, v = 30\ km/h$



(D) $r = 50\ m, v = 30\ km/h$

FIGURE 4.12: Average amount of overall daily energy saving for Configuration 3, as a function of network size, consumer speed, and application settings.

Conclusions

This thesis work aimed to design and analyze a novel protocol architecture in the context of the Future Internet. More in detail, it focused on finding a solution to the drawbacks of Information-Centric Networking related to the consumer mobility by embracing Software-Defined Networking and Multi-access Edge Computing.

To this end, this work analyzed the current state of the art on Information-Centric Networking, Software-Defined Networking, and Multi-access Edge Computing, by posing particular attention on the contributions focused on consumer mobility and highlighting the main drawbacks of information-centric paradigms (e.g., Named-Data Networking). First, it presented a preliminary solution, based on Software-Defined Networking, to the main drawbacks of Named-Data Networking. Specifically, it proposed a protocol architecture for dynamically updating forwarding functionalities through the control plane when the consumer detaches from the network and restoring the synchronization between consumer and producer when the former one attaches to a new network attachment point. Then, this thesis work envisioned an advanced protocol architecture that successfully integrates and properly customizes the key functionalities of Information-Centric Networking, Multi-access Edge Computing, and Software Defined Networking paradigms, in order to address consumer mobility, improve network performance and guarantee a better (and fully information-centric) management of network control operations.

The study on the impact of the conceived protocol architecture included several KPIs, including average communication overhead on the data plane, average communication overhead on the control plane, average overhead reduction achieved with respect to reference approaches exploiting the baseline pull-based strategy, as well as the bandwidth, memory, and energy savings obtained by deleting wrong forwarding information in intermediary network routers. Obtained results demonstrate that the proposed approach achieves a reduction of the communication overhead, that is confirmed in different topology, mobility, and application settings, also in presence of multiple consumers. The analysis also proved that the adoption of information-centric communication primitives for the control plane ensures an overhead reduction with respect to an implementation based on the conventional OpenFlow protocol. Only in extreme scenarios with a small network size, low traffic load, and very high number of mobile consumers asking for the same real-time contents, the amount of bandwidth consumed on the control plane appears comparable with the waste of bandwidth avoided on the data plane.

Future research activities) This thesis work already highlighted the capabilities of the proposed protocol architecture to reduce bandwidth, memory, and energy waste in case of consumer mobility, but future research activities will extend the conducted analysis on several aspects.

First, the future work will deeply analyze the behavior of the proposed solution on a different set of key performance indexes (e.g., latency, throughput), number of consumers/producers, and mobility models. This will further investigate the capabilities of the proposal and add detail to the impact of the proposal on the network.

Second, the implementation of the proposed protocol architecture will embrace the design of the next generation software-defined architectures (i.e., NG-SDN) and include other novel protocols for the southbound interface (e.g., P4/XDP and CAPWAP). As a matter of fact, NG-SDN represents the next step in programmable network architectures and leverages the latest softwarization technologies.

Third, the proposal will be integrated into a flexible architecture using Pervasive Intelligence (for instance, in 6G) to exploit mobility prediction when it is available and reduce the time required to delete the wrong forwarding information from the network.

Fourth, future analysis will also include the deployment of an experimental testbed to further describe the behavior of the proposed protocol architecture on real hardware devices.

Finally, the proposed protocol architecture will be extended to include a solution for the drawbacks of NDN related to producer mobility.

Bibliography

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," in *proc. of ACM International conference on Emerging networking experiments and technologies*, 2009, pp. 1–12.

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

[3] Z. Li, Y. Xu, B. Zhang, L. Yan, and K. Liu, "Packet Forwarding in Named Data Networking Requirements and Survey of Solutions," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1950–1987, 2018.

[4] A. M. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang, "Nlsr: Named-data link state routing protocol," in *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*, 2013, pp. 15–20.

[5] A. Detti, D. Tassetto, N. B. Melazzi, and F. Fedi, "Exploiting content centric networking to develop topic-based, publish–subscribe MANET systems," *Ad hoc networks*, vol. 24, pp. 115–133, 2015.

[6] W. Shang, A. Bannis, T. Liang, Z. Wang, Y. Yu, A. Afanasyev, J. Thompson, J. Burke, B. Zhang, and L. Zhang, "Named data networking of things," in *2016 IEEE first international conference on internet-of-things design and implementation (IoTDI)*, IEEE, 2016, pp. 117–128.

[7] C. Tsilopoulos and G. Xylomenos, "Supporting Diverse Traffic Types in Information Centric Networks," in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, ACM, 2011, pp. 13–18.

[8] J. François, T. Cholez, and T. Engel, "Ccn traffic optimization for iot," in *2013 Fourth international conference on the network of the future (NOF)*, IEEE, 2013, pp. 1–5.

[9] M. Amadeo, C. Campolo, and A. Molinaro, "Internet of things via named data networking: The support of push traffic," in *2014 International Conference and Workshop on the Network of the Future (NOF)*, IEEE, 2014, pp. 1–5.

[10] J. Chen, M. Arumaithurai, L. Jiao, X. Fu, and K. Ramakrishnan, "COPSS: An Efficient Content Oriented Publish/Subscribe System," in *proc. of IEEE Seventh ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, 2011, pp. 99–110.

[11] W. Drira and F. Filali, "A Pub/Sub extension to NDN for efficient data collection and dissemination in V2X networks," in *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, IEEE, 2014, pp. 1–7.

[12] A. V. Ventrella, G. Piro, and L. A. Grieco, "Publish-subscribe in mobile information centric networks: Modeling and performance evaluation," *Computer Networks*, vol. 127, pp. 317–339, 2017.

[13] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[14] D. Wu, X. Nie, E. Asmare, D. Arkhipov, Z. Qin, R. Li, J. McCann, and K. Li, "Towards distributed SDN: Mobility management and flow scheduling in software defined urban IOT," *IEEE Transactions on Parallel and Distributed Systems*, 2018.

[15] R. T. Cordova, P. R. Gondim, Y. P. Llerena, and J. Lloret, "SDN-DMM for intelligent mobility management in heterogeneous mobile IP networks," *International Journal of Communication Systems*, e4140, 2019.

[16] C.-M. Huang, D.-T. Dao, and M.-S. Chiang, "SDN-FHOR-DMM: a software defined network (SDN)-based fast handover with the optimal routing control method for distributed mobility management (DMM)," *Telecommunication Systems*, pp. 1–21, 2019.

[17] P. Palagummi and K. M. Sivalingam, "Smartho: A network initiated handover in ng-ran using p4-based switches," in *2018 14th International Conference on Network and Service Management (CNSM)*, IEEE, 2018, pp. 338–342.

[18] G. Siracusano, S. Salsano, P. L. Ventre, A. Detti, O Rashed, and N. Blefari-Melazzi, “A framework for experimenting ICN over SDN solutions using physical and virtual testbeds,” *Computer Networks*, vol. 134, pp. 245–259, 2018.

[19] S. Rowshanrad, M. R. Parsaei, and M. Keshtgari, “Implementing NDN using SDN: a review on methods and applications,” *IIUM Engineering Journal*, vol. 17, no. 2, pp. 11–20, 2016.

[20] H. L. Mai, M. Aouadj, G. Doyen, W. Mallouli, E. M. de Oca, and O. Festor, “Toward Content-Oriented Orchestration: SDN and NFV as Enabling Technologies for NDN,” in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, IEEE, 2019, pp. 594–598.

[21] Q.-Y. Zhang, X.-W. Wang, M. Huang, K.-Q. Li, and S. K. Das, “Software defined networking meets information centric networking: A survey,” *IEEE Access*, vol. 6, pp. 39 547–39 563, 2018.

[22] N. L. van Adrichem and F. A. Kuipers, “NDNFlow: Software-defined named data networking,” in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, IEEE, 2015, pp. 1–5.

[23] J. Li, K. Xue, J. Liu, Y. Zhang, and Y. Fang, “An icn/sdn-based network architecture and efficient content retrieval for future satellite-terrestrial integrated networks,” *IEEE Network*, vol. 34, no. 1, pp. 188–195, 2019.

[24] T. Combe, W. Mallouli, T. Cholez, G. Doyen, B. Mathieu, and E. M. De Oca, “An sdn and nfv use case: Ndn implementation and security monitoring,” in *Guide to Security in SDN and NFV*, Springer, 2017, pp. 299–321.

[25] H. Zhang, R. Xie, S. Zhu, T. Huang, and Y. Liu, “Dena: An intelligent content discovery system used in named data networking,” *IEEE Access*, vol. 4, pp. 9093–9107, 2016.

[26] X. N. Nguyen, D. Saucez, and T. Turletti, “Efficient caching in content-centric networks using openflow,” in *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, 2013, pp. 67–68.

[27] A. Chanda, C. Westphal, and D. Raychaudhuri, “Content based traffic engineering in software defined information centric networks,” in *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, 2013, pp. 357–362.

[28] Q. Sun, W. Wendong, Y. Hu, X. Que, and G. Xiangyang, "Sdn-based autonomic ccn traffic management," in *2014 IEEE Globecom Workshops (GC Wkshps)*, IEEE, 2014, pp. 183–187.

[29] R. Huo, F. R. Yu, T. Huang, R. Xie, J. Liu, V. C. Leung, and Y. Liu, "Software defined networking, caching, and computing for green wireless networks," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 185–193, 2016.

[30] W.-S. Kim, S.-H. Chung, and J.-W. Moon, "Improved content management for information-centric networking in sdn-based wireless mesh network," *Computer Networks*, vol. 92, pp. 316–329, 2015.

[31] M. Chowdhury, A. Gawande, and L. Wang, "Secure information sharing among autonomous vehicles in ndn," in *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*, IEEE, 2017, pp. 15–26.

[32] W. Xia, J. Zhang, T. Q. Quek, S. Jin, and H. Zhu, "Mobile edge cloud-based industrial internet of things: Improving edge intelligence with hierarchical sdn controllers," *IEEE Vehicular Technology Magazine*, vol. 15, no. 1, pp. 36–45, 2020.

[33] Y. Zhang, L. Cui, W. Wang, and Y. Zhang, "A survey on software defined networking with multiple controllers," *Journal of Network and Computer Applications*, vol. 103, pp. 101–118, 2018.

[34] Open Networking Foundation, *NG-SDN*, (accessed July 12, 2020). [Online]. Available: `https://www.opennetworking.org/ng-sdn/`.

[35] M. Li, X. Wang, H. Tong, T. Liu, and Y. Tian, "SPARC: Towards a Scalable Distributed Control Plane Architecture for Protocol-Oblivious SDN Networks," in *2019 28th International Conference on Computer Communication and Networks (ICCCN)*, IEEE, 2019, pp. 1–9.

[36] Z. Jia, J. Wang, X. Chen, and L. Kang, "Enable Device-Aware Flow Control with Enhanced Protocol-Oblivious Forwarding (POF)," in *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, IEEE, 2019, pp. 5–8.

[37] S. Li, D. Hu, W. Fang, S. Ma, C. Chen, H. Huang, and Z. Zhu, "Protocol oblivious forwarding (POF): Software-defined networking with enhanced programmability," *IEEE Network*, vol. 31, no. 2, pp. 58–66, 2017.

[38] P. Gallo, K. Kosek-Szott, S. Szott, and I. Tinnirello, "Cadwan: A control architecture for dense wifi access networks," *IEEE communications magazine*, vol. 56, no. 1, pp. 194–201, 2018.

[39] A. S. Mubarak, H. Esmaiel, and E. M. Mohamed, "New capwap architectures for ieee 802.11 ad based wi-fi/wigig wlans," in *2018 International Conference on Innovative Trends in Computer Engineering (ITCE)*, IEEE, 2018, pp. 231–235.

[40] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[41] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[42] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2017.

[43] Z. Rejiba, X. Masip-Bruin, and E. Marín-Tordera, "A Survey on Mobility-Induced Service Migration in the Fog, Edge, and Related Computing Paradigms," *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, p. 90, 2019.

[44] P. Zhang, M. Durresi, and A. Durresi, "Multi-access edge computing aided mobility for privacy protection in Internet of Things," *Computing*, vol. 101, no. 7, pp. 729–742, 2019.

[45] C.-M. Huang, D.-T. Dao, and M.-S. Chiang, "A MEC-Assisted Method for Early Handover Using the Fully Distributed Mobility Management (MEC-F-DMM) Architecture," in *International Conference on Advanced Information Networking and Applications*, Springer, 2019, pp. 737–750.

[46] T. Ojima and T. Fujii, "Resource Management for Mobile Edge Computing using User Mobility Prediction," in *2018 International Conference on Information Networking (ICOIN)*, IEEE, 2018, pp. 718–720.

[47] M. Amadeo, C. Campolo, A. Molinaro, C. Rottondi, and G. Verticale, "Securing the Mobile Edge through Named Data Networking," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, IEEE, 2018, pp. 80–85.

[48] P. Hao and X. Wang, "Integrating PHY Security Into NDN-IoT Networks By Exploiting MEC: Authentication Efficiency, Robustness, and Accuracy Enhancement," *arXiv preprint arXiv:1904.03283*, 2019.

[49] D. Grewe, M. Wagner, M. Arumaithurai, I. Psaras, and D. Kutscher, "Information-centric mobile edge computing for connected vehicle environments: Challenges and research directions," in *Proceedings of the Workshop on Mobile Edge Communications*, ACM, 2017, pp. 7–12.

[50] X. Sun and N. Ansari, "EdgeIoT: Mobile edge computing for the Internet of Things," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 22–29, 2016.

[51] A. C. Baktir, A. Ozgovde, and C. Ersoy, "How can edge computing benefit from software-defined networking: A survey, use cases, and future directions," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2359–2391, 2017.

[52] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song, and M. Qiu, "A Scalable and Quick-Response Software Defined Vehicular Network Assisted by Mobile Edge Computing," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 94–100, 2017.

[53] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.

[54] Y. Ai, M. Peng, and K. Zhang, "Edge computing technologies for Internet of Things: a primer," *Digital Communications and Networks*, vol. 4, no. 2, pp. 77–86, 2018.

[55] V. Balasubramanian, F. Zaman, M. Aloqaily, I. Al Ridhawi, Y. Jararweh, and H. B. Salameh, "A mobility management architecture for seamless delivery of 5G-IoT services," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, IEEE, 2019, pp. 1–7.

[56] R. Ravindran, X. Liu, A. Chakraborti, X. Zhang, and G. Wang, "Towards software defined icn based edge-cloud services," in *2013 IEEE 2nd International Conference on Cloud Networking (CloudNet)*, IEEE, 2013, pp. 227–235.

[57] M. Amadeo, C. Campolo, G. Ruggeri, A. Molinaro, and A. Iera, "Towards Software-defined Fog Computing via Named Data Networking," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, 2019, pp. 133–138.

[58] ——, "Sdn-managed provisioning of named computing services in edge infrastructures," *IEEE Transactions on Network and Service Management*, vol. 16, no. 4, pp. 1464–1478, 2019.

[59] H. Li, K. Ota, and M. Dong, "Eccn: Orchestration of edge-centric computing and content-centric networking in the 5g radio access network," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 88–93, 2018.

[60] S. H. Ahmed, S. H. Bouk, D. Kim, D. B. Rawat, and H. Song, "Named Data Networking for Software Defined Vehicular Networks," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 60–66, 2017.

[61] E. Kalogeiton, Z. Zhao, and T. Braun, "Is SDN the solution for NDN-VANETs?" In *2017 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, IEEE, 2017, pp. 1–6.

[62] J. Ge, S. Wang, Y. Wu, H. Tang, and E Yuepeng, "Performance improvement for source mobility in named data networking based on global–local FIB updates," *Peer-to-Peer Networking and Applications*, vol. 9, no. 4, pp. 670–680, 2016.

[63] Y. Zhang, A. Afanasyev, J. Burke, and L. Zhang, "A Survey of Mobility Support in Named Data Networking," in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, 2016, pp. 83–88.

[64] B. Feng, H. Zhou, and Q. Xu, "Mobility support in named data networking: A survey," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, p. 220, 2016.

[65] B. Nour, H. I. Khedher, H. Moungla, H. Afifi, F. Li, S. Kashif, H. Khelifi, and M. Guizani, "Internet of Things Mobility over Information-Centric/Named-Data Networking," *IEEE Internet Computing*, 2019.

[66] J.-H. Cha, J.-H. Choi, J.-Y. Kim, Y.-H. Han, and S.-G. Min, "A Mobility Link Service for NDN Consumer Mobility," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[67] Z. Yan, G. Geng, S. Zeadally, and Y.-J. Park, "Distributed all-ip mobility management architecture supported by the NDN overlay," *IEEE Access*, vol. 5, pp. 243–251, 2016.

[68] S. Adhatarao, M. Arumaithurai, D. Kutscher, and X. Fu, “NeMoI: Network mobility in ICN,” in *International Conference on Communication Systems and Networks*, Springer, 2018, pp. 220–244.

[69] A. Azgin, R. Ravindran, and G. Wang, “On-demand mobility support with anchor chains in Information Centric Networks,” in *2017 IEEE International Conference on Communications (ICC)*, IEEE, 2017, pp. 1–7.

[70] X. Wang and Y. Li, “Mobility management in content-centric networking,” *Telecommunication Systems*, pp. 1–16, 2019.

[71] R. A. Rehman and B.-S. Kim, “LOMCF: Forwarding and caching in named data networking based MANETs,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 9350–9364, 2017.

[72] M. F. Majeed, S. H. Ahmed, and M. N. Dailey, “Enabling Push-Based Critical Data Forwarding in Vehicular Named Data Networks,” *IEEE Communications Letters*, vol. 21, no. 4, pp. 873–876, 2016.

[73] L. Rui, S. Yang, and H. Huang, “A proactive multi-level cache selection scheme to enhance consumer mobility support in Named Data Networking,” *International Journal of Distributed Sensor Networks*, vol. 13, no. 4, p. 1 550 147 717 700 897, 2017.

[74] H. Khelifi, S. Luo, B. Nour, A. Sellami, H. Moungla, and F. Naït-Abdesselam, “An Optimized Proactive Caching Scheme based on Mobility Prediction for Vehicular Networks,” in *2018 IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2018, pp. 1–6.

[75] H. Farahat and H. S. Hassanein, “Supporting Consumer Mobility Using Proactive Caching in Named Data Networks,” in *2016 IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2016, pp. 1–6.

[76] A. Azgin, R. Ravindran, and G. Wang, “Mobility study for named data networking in wireless access networks,” in *2014 IEEE International Conference on Communications (ICC)*, IEEE, 2014, pp. 3252–3257.

[77] C. Fang, H. Yao, Z. Wang, W. Wu, X. Jin, and F. R. Yu, “A survey of mobile information-centric networking: Research issues and challenges,” *IEEE Communications Surveys & Tutorials*, 2018.

[78] L. Liu, Z. Ye, and A. Ito, “CAMS: coordinator assisted mobility support for seamless and bandwidth-efficient handover in ICN,” in *proc. of IEEE Globecom Workshops (GC Wkshps)*, 2015, pp. 1–7.

[79] F. Ren, Y. Qin, H. Zhou, and Y. Xu, "Mobility management scheme based on Software Defined Controller for Content-Centric Networking," in *proc. of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2016, pp. 193–198.

[80] R. Soua, E. Kalogeiton, G. Manzo, J. M. Duarte, M. R. Palattella, A. Di Maio, T. Braun, T. Engel, L. A. Villas, and G. A. Rizzo, "SDN coordination for CCN and FC content dissemination in VANETs," in *Ad Hoc Networks*, Springer, 2017, pp. 221–233.

[81] J. M. Duarte, E. Kalogeiton, R. Soua, G. Manzo, M. R. Palattella, A. Di Maio, T. Braun, T. Engel, L. A. Villas, and G. A. Rizzo, "A Multi-Pronged Approach to Adaptive and Context Aware Content Dissemination in VANETs," *Mobile Networks and Applications*, vol. 23, no. 5, pp. 1247–1259, 2018.

[82] E. Schiller, N. Nikaein, E. Kalogeiton, M. Gasparyan, and T. Braun, "CDS-MEC: NFV/SDN-based application management for MEC in 5G systems," *Computer Networks*, vol. 135, pp. 96–107, 2018.

[83] F. Wu, W. Yang, R. Chen, and X. Xie, "Broadband communications for high-speed trains via ndn wireless mesh network," *Tsinghua Science and Technology*, vol. 23, no. 4, pp. 419–430, 2018.

[84] K. Kanai, T. Muto, J. Katto, S. Yamamura, T. Furutono, T. Saito, H. Mikami, K. Kusachi, T. Tsuda, W. Kameyama, *et al.*, "Proactive Content Caching for Mobile Video Utilizing Transportation Systems and Evaluation Through Field Experiments," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2102–2114, 2016.

[85] S. Shailendra, B. Panigrahi, H. K. Rath, and A. Simha, "A novel overlay architecture for information centric networking," in *2015 Twenty First National Conference on Communications (NCC)*, IEEE, 2015, pp. 1–6.

[86] L. Wang, A. Hoque, C. Yi, A. Alyyan, and B. Zhang, "Ospfn: An ospf based routing protocol for named data networking," Technical Report NDN-0003, Tech. Rep., 2012.

[87] A.-L. Barabási *et al.*, *Network science*. Cambridge university press, 2016.

[88] B. Bollobás and O. Riordan, "The diameter of a scale-free random graph," *Combinatorica*, vol. 24, no. 1, pp. 5–34, 2004.

[89] K. L. Yeung and S. Nanda, "Channel management in microcell/-macrocell cellular radio systems," *IEEE Transactions on Vehicular Technology*, vol. 45, no. 4, pp. 601–612, 1996.

[90] A. Medina, A. Lakhina, I. Matta, and J. Byers, "Brite: An approach to universal topology generation," in *MASCOTS 2001, Proceedings Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, IEEE, 2001, pp. 346–353.

[91] M. Alhowaidi, B. Ramamurthy, B. Bockelman, and D. Swanson, "SDTMA-NDN: Scientific data transfer management architecture using named data networking," in *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, IEEE, 2017, pp. 1–6.

[92] H. Lim, A. Ni, D. Kim, Y.-B. Ko, S. Shannigrahi, and C. Papadopoulos, "NDN construction for big science: lessons learned from establishing a testbed," *IEEE Network*, vol. 32, no. 6, pp. 124–136, 2018.

[93] B. Han, X. Wang, N. Choi, T. Kwon, and Y. Choi, "AMVS-NDN: Adaptive mobile video streaming and sharing in wireless named data networking," in *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, 2013, pp. 375–380.

[94] 3GPP, "Technical Specification Group Services and System Aspects; Service requirements for cyber-physical control applications in vertical domains; Stage 1," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 22.104, Dec. 2019, Version 17.2.0. [Online]. Available: `https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2440`.

[95] V. Jacobson, J. Burke, L. Zhang, B. Zhang, K. Claffy, D. Krioukov, C. Papadopoulos, L Wang, E Yeh, and P Crowley, "Named data networking (NDN) project 2013-2014 report," *Palo Alto Research Centre*, 2014.

[96] K. T. Ko, H. H. Hlaing, and M. Mambo, "A peks-based ndn strategy for name privacy," *Future Internet*, vol. 12, no. 8, p. 130, 2020.

[97] H. Im and D. Kim, "An overview of content poisoning in ndn: Attacks, countermeasures, and direction," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 14, no. 7, pp. 2904–2918, 2020.

[98] D. FANTASTIC5G, "D2. 1 Air Interface Framework and Specification of System Level Simulations," *Call: H2020-ICT-2014-2, Project reference*, vol. 671660, 2016.

[99] N. Patriciello, J. Núñez-Martínez, J. Baranda, M. Casoni, and J. Mangues-Bafalluy, "Tcp performance evaluation over backpressure-based routing strategies for wireless mesh backhaul in lte networks," *Ad Hoc Networks*, vol. 60, pp. 40–51, 2017.

[100] T. Al-Mudaihesh, S. Alyahya, and M. M. Hassan, "A publish-subscribe model for cloud-based Internet Protocol television surveillance systems," in *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, IEEE, 2016, pp. 955–960.

[101] K. Cabaj, M. Gregorczyk, and W. Mazurczyk, "Software-defined networking-based crypto ransomware detection using http traffic characteristics," *Computers & Electrical Engineering*, vol. 66, pp. 353–368, 2018.

[102] A. Afanasyev, C. Yi, L. Wang, B. Zhang, and L. Zhang, "SNAMP: Secure namespace mapping to scale NDN forwarding," in *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, 2015, pp. 281–286.

[103] Y. Wang, B. Xu, D. Tai, J. Lu, T. Zhang, H. Dai, B. Zhang, and B. Liu, "Fast Name Lookup for Named Data Networking," in *2014 IEEE 22nd International Symposium of Quality of Service (IWQoS)*, IEEE, 2014, pp. 198–207.

[104] OpenFlow Switch Specification, *Version 1.5.1 (protocol version 0x06)*, 2015.

[105] Y. Ding, X. Li, Y.-C. Tian, G. Ledwich, Y. Mishra, and C. Zhou, "Generating scale-free topology for wireless neighborhood area networks in smart grid," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 4245–4252, 2018.

[106] H. Dai, Y. Wang, J. Fan, and B. Liu, "Mitigate DDoS attacks in NDN by interest traceback," in *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, 2013, pp. 381–386.

[107] K. Ohsugi, J. Takemasa, Y. Koizumi, T. Hasegawa, and I. Psaras, "Power consumption model of ndn-based multicore software router based on detailed protocol analysis," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1631–1644, 2016.