University of Padua
Human Inspired Technology Research Centre
Curriculum: Computer Science and Innovation for Societal Challenges

Securing Information Centric Networking

Candidate
Muhammad Hassan
Raza Khan

Supervisor
Prof. Mauro Conti

Co-Supervisor
Prof. Anna Spagnolli

University of Padova, Italy

May 14, 2019

Acknowledgments

The work presented here was carried out as part of my doctoral studies at the University of Padova and it would not have been possible without the motivation, guidance and support of many people to whom I would like to express my gratitude.

Foremost, my deepest gratitude goes to my supervisor Prof. Mauro Conti for guiding me towards the achievement of the doctoral degree and giving me the opportunity to join the SPRITZ Security and Privacy Research Group of University of Padova. I have been extremely lucky to have his continuous support and most importantly Mauro also guided me how to become an independent and mature researcher.

Second, I would like to thank my co-advisor Prof. Anna Spagnolli for her support and guidance, especially in scientific writing and how to improve my research towards human-need. I also would like to thank Dr. Ralph Droms (Google Cambridge U.S.A), Prof. Thorstern Sturfe (Technical University Dresden, Germany), Dr. Chaggan Lal (University of Padova, Italy) and Dr. Alberto Compagno (Cisco Systems, France) for their valuable guidance and contribution throughput my research activities.

I would like to thank my friends Pallavi Kaliyar and Masoom Rabbani, all colleagues and the members of SPRITZ group, especially for all those stimulating discussions, arguments and fun which greatly helped me in my research.

Last but not least, a special thank goes to my beloved friend and family, who always supported me in these years and motivated me to achieve my goals.

Muhammad Hassan Raza Khan
Padova, May 14, 2019

Abstract

In recent years, the usage model of the current Internet has experienced an unexpected paradigm shift due to overwhelming requirements in distributed content distribution, device mobility, network scalability, information retrieval, network-based services to name a few. To address these pressing requirements, along with inherent security (which was completely ignored by the early adopters of the Internet), researchers have proposed the Information Centric Networking (ICN) paradigm as a possible replacement of the current host-centric communication model.

In ICN, named content turns out to be a "first-class entity", thus focusing on efficient content distribution, which is debatably not well served by current Internet. Several projects have embraced the ICN philosophy and aim at proposing a viable future Internet architecture. However, to successfully accomplish the objective, ICN and its implementing projects should include a leading obligation in their design, i.e., support security from the outset. To evade the prolonged and endured past of incremental security-patching and retrofitting that characterizes the current Internet architecture. This dissertation goes into such direction and focuses on securing ICN paradigm and its implementing architectures. In particular, this dissertation contributes by: (i) addressing vulnerabilities in the interaction of ICN's implicit features and widely used existing technology, such as multimedia streaming; (ii) exploiting ICN intrinsic mobility support to provide security services to the upper layer, such as authentication; (iii) securing the ICN mobility features; (iv) addressing the architectural security issues that are intrinsic to the ICN design; and (v) addressing the feasibility and effectiveness of ICN with respect to real world deployment configurations.

In the first part of this dissertation, we focus on secure and efficient multimedia streaming in ICN. In particular, we identify a novel vulnerability that

adversely exploits two fundamental ICN characteristics: in-network caching and multicast support. We show that an adversary with limited resources is able to disrupt the adaptive behaviour of multimedia streaming control system to degrade the perceived Quality of Experience (QoE) of a benign user. To address the problem, we proposed two counter approaches. First, we propose a user driven approach called Fair-RTT-DAS to countermeasure the attack. Through extensive simulations, we show that Fair-RTT-DAS ensures efficient bandwidth utilization and significantly alleviates the adverse effects. In the second approach, we aim to eliminate the deficiencies of ICN architectural features related to caching and forwarding. We implement coordination with lightweight monitoring for efficient multimedia streaming to enable network-wide coordinated caching and cache-aware routing. By this, it aims to reduce quality fluctuations and cache content redundancy in presence of both adversary and inherent content source variations, thus it enhances the perceived QoE. Extensive simulation studies show the effectiveness and feasibility of proposed approaches.

In the second part of this thesis, we revise the mobility management and propose an simplified Long Term Evolution (LTE) infrastructure that exploits the mobility support provided at the ICN network layer. We revamp the current device authentication protocol for LTE, and we present a novel handover protocol that exploits the ICN communication style. Compared to the protocol adopted in the current LTE, our proposals are able to reduce the number of messages required to authenticate or re-authenticate a device during mobility.

In the third part of this thesis, we consider the fundamental security issues related to the mobility management protocols in ICN. We identify that installing the current mobility solutions lack an adequate security mechanism and invites severe security threats, e.g., prefix hijacking, Denial of Service (DoS) attacks, in the network. To address these security threats, we propose a Blockchain based lightweight distributed mobile producer Authentication (BlockA*uth*) protocol to enable secure and efficient mobility management in ICN. We present a qualitative security analysis which confirms robustness of BlockA*uth* against a wide array of security attacks to which mobile network and blockchain are particularly vulnerable such as prefix hijacking, DoS attacks. In addition, the performance evaluation of BlockA*uth* shows that it maintains significant performance gain compared to the state-of-the-art prefix attestation proposals.

In the fourth part of this thesis, we propose a novel prevention mechanism for a specific type of DoS attack, i.e., Interest Flooding Attack (IFA), which aims to reduce the network's memory resource consumption and tar-

gets malicious traffic only. In particular, the protocol exploits a decent queue management strategy to avoid congestion problem and in consequence attacks. To evaluate the effectiveness of our work, we implement the IFA and proposed protocol on the simulated environment. The results show that proposed protocol effectively prevents from the adverse effects of IFA and shows significantly less false positives as compared to the state of the art IFA mitigation approaches.

In the last part of this thesis, we take into account the importance of the transition phase during which current and future Internet architectures will coexist. Various research projects have addressed the coexistence of IP and ICN following various techniques. Our contribution under this part aims at providing a comprehensive analysis, classification and comparison of the coexistence architectures according to their features (i.e., deployment approach, deployment scenarios, architecture or technology used and addressed coexistence requirements) and evaluation criteria (i.e., challenges emerging during the deployment and the runtime behaviour of an architecture). We believe that this work will finally fill the gap required for moving towards the design of the final coexistence architecture.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

History of Communication

Communication is an element of instinctive humanoid behaviour that enables transmission of data in a one-to-one, one-to-many, or many-to-many styles. Communication among humans begun with vocal messages. Therefore, strong physical proximity was required which possibly necessitate long-distance journey and/or implicit communication (i.e., through mediators). In the beginning, to overwhelm these constraints of vocal communication, messengers, pigeons, drums and smoke signals remained in by many civilizations. This characterized the early phases of telecommunication.

It happened after the early 19th century that electrical telecommunication systems emerged. The electrical telegraphy acclaimed as a groundbreaking discovery compared to its electromagnetic equivalent [39]. Later in the same era, installation of first successful trans-atlantic cable came to existence, providing telecommunication between North America and Europe. In the late 19th century, Alexander Graham Bell laid the foundation of audio telecommunication by inventing telephone [91]. This helped to launch the first commercial audio telecommunication facilities throughout the Atlantic. Later, the invention followed the implementation of telephone networks controlled via human operators.

Initial study concerning wireless communication [85] set the foundation of radio systems - the leading devices capable to wirelessly transmit vocal messages. Through this, communication systems penetrated into the deployments scenarios which were more ad-hoc such as uninhabited or hostile environments. Nevertheless, the ambitions of society were not stopped just

by transmitting vocal messages over wireless channels. In the first half of the 20th century, the invention of television permitted the transmission of videos and images over the same channels.

The Internet of Today

Circuit-switched networking was founded by establishing a reserved channel between two communicating sides [118]. Initially, the technology was applied to transmit simple numerical data, and is still being in use to deploy today's telephone networks. Nevertheless, the scalability of circuit-switched networking was a severe challenge once it happened to transmit data over the network. Packet-switched networking was then introduced in the late 1960s, and at that time, believed to be more suitable for data transmission. In particular, in packet switching, data is divided into smaller packets to transmit between the computers over a medium which is shared with other computers.

In the late 1960s, The first implementation of packet switching the ARPANET [152], Advanced Research Projects Agency Network came to existence which was funded by the Department of Defense of the United States. In its initial phases, ARPANET used to support only a few applications in which the first was email. Even till the early 1970s, the majority of ARPANET traffic was comprised of email messages.

Later in 1981, two vital inventions of RFCs (Request for Comments) were published, named as the Internet Protocol v4 (IPv4) [8] and Transmission Control Protocol (TCP) [9]. Together, they laid the foundation for today's Internet by establishing the TCP/IP protocol suite. The IP protocol is responsible to route the packets, termed as datagrams[1], from source to destination host. In the view of IP, the Internet is considered as a collection of interconnected networks and Autonomous Systems (AS-s). In particular, on Internet each network entity, e.g., router or host, is recognized through a distinct IP address[2]. Furthermore, each IP address entails two parts: (i) the network prefix, and (ii) the host identifier. These specific design features force the IP addressing scheme to scale with the increasing number of network hosts. Another vital feature delivered by IPv4 is fragmentation. Whenever the size of an IP Packet is greater than the defined Maximum Transmission Unit (MTU) of forwarding interface, the packet is divided into smaller parts, known as fragments. Correspondingly, to recover the original packet, destination host reassemble the received fragments. Other network

[1] We denote the terms IP packet and IP datagram interchangeably.
[2] Generally IP addresses are assigned to the interfaces of an entity.

entities such as Network Address Translation Tables (NAT) [73] and in-network firewalls are sometimes also responsible to assemble fragments.

The Missing Piece in the Internet Design: Security

The longevity and popularity of IP were not foreseen from the time when it was invented, thus security and privacy were completely ignored and were not supported by design. IPsec [181] was then designed as an "add-on patch" in order to provide data origin authentication, integrity, and confidentiality of IP packets. In particular, the first two objectives are achieved through Authentication Header (AH) protocol [106], whereas Encapsulating Security Payload (ESP) protocol [107] intends to provide all three security features. IPsec entails two modes of operation:

- Transport mode functions to provide secure end-to-end communication, e.g., client-to-server communication. In particular, only the payload of the packet is encrypted and authenticated in transport mode. Transport and application layers packets are protected through hashing, thus they cannot be modified utilizing NAT. To overcome this issue, NAT-Traversal[3] [110] is deployed.

- Tunnel mode is generally applied among the gateways, in order to provide a secure connection between networks, e.g., various sites of the same company. In addition, tunnel mode is also used to deliver secure host-to-gateway communications. For instance, whole IP packets are encrypted and then encapsulated into new IP packets. A typical application of tunnel mode is Virtual Private Networks (VPN) [138].

The research and development of Internet have received an incredible speed-up, where everyone aims to show the potentiality of IP and TCP protocols by developing the applications (e.g., video transfer, wireless access) and enhancing the performance as much as possible. However, all this development based on the simple assumption that internauts would have used the Internet, and its applications, in the way how the Internet has been designed. This steered to design applications and protocols without considering the foundations of security. Although the above-mentioned add-on security patches tried their utmost to meet security goals, researchers discovered the consequences of their mistaken assumption and inattention. One classic example which seconds the mistaken assumption was *Morris worm* [157], which was the first disruptive worm highlighting the security

[3]NAT-T is an approach to solve IP address translation problems which are encountered when data protected by IPsec passes through a NAT device for address translation.

flaws released by a student from the Massachusetts Institute of Technology in 1988. Later, it opens a thriving era for hackers and malicious intenders who showed plenty of techniques to breach security primitives.

The Progression around Internet and its Implications

Applications and network technology, both have exhibited a tremendous advancement which started from the very first design of ARPANET and still on its way. The initial slow wired technology, offered at the time of ARPNET, has now been replaced with extremely fast home-to-home fiber connections and ubiquitous wireless technologies which are able to provide a bandwidth of the order of tens of megabits per seconds. Compared to the initial application of Internet, a straightforward resource sharing system [193], existing applications offers whole new set of services (e.g., online video streaming, audio/video sharing, real-time video conferencing, TV, radio broadcasting and high-speed wireless connectivity) which are light years away from the initial Internet application. However, compared to existing applications and technology, IP and TCP protocols have not been subject to the similar substantial transformation, demonstrating all their agility and decent design in endorsing the above (application) and below (network technology) changes from many years. The modern inventions in network technologies (e.g., mobile networks) and applications (e.g., video streaming services, news, images) which have shaken the way people use the Internet are:

- The latest and most popular applications focus primarily on sharing and accessing multimedia content (e.g. video streaming in real time, online gaming, social media sharing). These applications influenced the behavior of the internet user and turned them into internauts, which shifted from reaching to the particular host and using its limited resources to share and retrieve content all over [209].

- Innovative mobile technologies (e.g., 4G, LTE, WiFi) make people to access Internet primarily from their mobile/wireless devices instead of a static desktop position.

Following are the fundamental issues and requirements of existing Internet, laid by the internaut's behaviours, which have most recently uncovered all the limitations of the current host-based communication of TCP/IP:

- Content retrieval. In the current IP and TCP based Internet, content retrieval is not an efficient and fast process. It mandates the

clients to know not only what is the content, but also where the requested content is located in the network, i.e., the IP address of the source which carries a copy of the content. This requires an additional procedure which aims to facilitate client in order to resolve the content name into an IP address (location), e.g., the Domain Name System (DNS) [10, 11]. In addition, the sources should also know the IP address of the client to which they can send each requested content. Correspondingly, the network routers forward the content and requests based on the IP address of the destination node. Nevertheless, content retrieval in the existing Internet is not aligned with the behaviour of internauts, where clients are more involved in the characteristics of content instead of where it is situated.

- Mobility. One obvious failure of IP is mobility management. Initially designed for the existing static wired network technology, IP failed to chase the technology evolution which allows nowadays wireless connectivity and mobile devices. Different research efforts [63, 163, 200] tried to overcome the lack of mobility in the IP design, however, none of them was really able to provide a cost-effective mobility mechanism. For this reason, mobility management is nowadays provided at link-layer, enabled only for specific wireless technologies (e.g., LTE, Wi-Fi) and confined in singular networks.

- Security. The notion of security was absolutely neglected in the fundamental design of IP and TCP. Afterwards, with a set of add-on "patches" and changes, now IP can assure integrity and confidentiality of communication between two, or several, particular hosts. Nevertheless, while communications are today based on the host-centric model, the security services of IP place trust in hosts. Thus binding the authenticity of the content associated with the trust in the host, i.e., from which the content has been retrieved.

Given the increasing number of limitations of the current Internet, researchers started designing new architectures, that aims to replace the current one in the close future [61, 67, 101, 112, 182, 222]. Among those, the most promising ones adhere to Information Centric Networking (ICN) [214]: a new network communication model in which the traditional host-centric paradigm has been moved to the new information-centric one. While in the current Internet two endpoints can start communicating only if they know the respective IP addresses, in ICN they can send requests specifying only the content names, without being aware of the content's location in the net-

work. In particular, ICN aims at replacing the current IP's network layer, and we realized from the past that this is the fundamental obligatory requirement to accomplish before deploying the new network technologies. In this thesis, we focus on security and access control issues of the architectures implementing the ICN paradigm.

In the following section, we present the ICN paradigm along with the state of the art of its architectural design.

1.1 Information Centric Networking Paradigm

The fundamental idea behind Information Centric Networking (ICN) paradigm is that *who* is communicating is far less important in communication than *what* is data is required. This major shift in communication paradigm has occurred due to the end-users use of todays Internet, which indeed is more content-centric than location-centric, e.g., video sharing, social networking, retrieving aggregated data. ICN style fundamentally decouples the content from its sources, through a clear location-identity split. The methodological assumption behind is that content should be named, directed and matched independently of its location since it may be present anywhere in the network. Therefore, instead of specifying a source-to-destination host pair for communication, ICN names the piece of content, making content (data) as a first-class entity.

The first practical implementation of ICN concept was introduced in 2001 as a guiding principle of the TRIAD project [46]. TRIAD, a project carried out by the University of Stanford, introduced a new *content layer* to the communication model, which provided several content-based features such as hierarchical content caching, content replication and its discovery, multicast-based content distribution and name-based routing. These features specify to use content names instead of addresses for routing. However, IP and TCP were still the basis of proposed architecture over which the content layer passes. In 2006, UC Berkeley and ICSI proposed the successor of TRIAD, the data-oriented network architecture (DONA) [112] which improved TRIAD by incorporating data authenticity and persistence as the key objectives of the architecture, but following the similar dependency of underlying IP and TCP. In 2009, Palo Alto Research Center (PARC) revealed Content Centric Network (CCN) [101] project which was led by the research fellow Van Jacobson. Soon after that, the National Science Foundation (NSF), inspired by the ICN, introduced its Future Internet Architecture program, which gave birth to the promising architecture, Named-Data Networking (NDN) [222], a branch of the CCN project. Both CCN and NDN

significantly move forward the TRIAD and DONA projects, i.e., introducing a new network layer with the aim of replacing the existing TCP and IP. CCN and NDN are two key projects which have gained considerable attention in the research groups of both academia and industry, influencing the drafting of the Information Centric Networking architecture [4].

Besides being prominent, the architectures of NDN and CCN are very similar, which make them almost indistinguishable for the scope of this thesis. Therefore, although our experimental evaluation is mostly based on the NDN Forwarding Daemon, to avoid any ambiguities, we refer to the reference architectures comprehensively as ICN in the rest of our thesis. In the few cases where the differences are meaningful for the subject we are discussing, we directly refer to NDN or CCN, stating why the subject does not apply (or we cannot guarantee that applies) to the other architecture. In the following, we provide a brief summary of the common design between NDN and CCN, namely the ICN architecture we consider.

1.2 ICN Architecture

The architecture of ICN follows the same hourglass model of TCP/IP, where the network layer stands as a thin-waist of the architecture, supplying the minimal functionality required for global inter-connectivity. In ICN, the network layer makes use of hierarchically structured names to directly address the content. In particular, names are comprised of human-readable components, e.g., `/example.com/video4u/examples.mp3` [222], where "/" defines the boundary between the name components. For the network, names are considered opaque, and for this purpose, it can also consist of binary components which if required can be converted to human-readable form while presenting to them the users.

To support efficient content distribution, ICN defines two types of packets: *interest* and *content* (the later is also denoted as data packet). ICN communication model can be characterized as using a *pull model*: content is delivered to consumers only upon (prior) explicit requests for that content, i.e., each content delivery is triggered by a request for that content. Content is generated by the producers which are also responsible for announcing its availability to the network. After receiving an interest, an ICN router forwards the interest towards the content producer responsible for the requested name, using longest name-prefix matching as routing information. Then, after the interest is delivered to the content producer, the producer responds by sending the content into the network, thus satisfying the in-

terest. The requested content packet is forwarded towards the consumer, traversing - in reverse - the path of the preceding interest.

ICN gives more responsibilities to the routers as it introduces router-side *content caching* and *interest aggregation* [101,222]. Through *in-network caching*, routers able to store recently received contents for the subsequent requests. Thus, an ICN interest might be satisfied by an actual content producer or by any intermediate router, providing native support of caching and content distribution at the network layer. Along with that *interest aggregation* features native support for multicast, i.e., only the first of multiple closely spaced (and timed) interests requesting the same content is forwarded upstream by each router. In order to implement the communication, each ICN network entity maintains the following three components:

- Content Store: is the cache used for content caching and retrieval. A routers cache size is determined by local resource availability. Each router unilaterally determines what content to cache and for how long. From here on, we use the terms CS and cache interchangeably.
- Forwarding Information Base (FIB): is the table of name prefixes and corresponding outgoing interfaces. FIB is used to route interests based on longest-prefix matching of their names.
- Pending Interest Table (PIT): is the table of outstanding (pending) interest names and a set of corresponding incoming interfaces, denoted as arrival-interfaces.

Caching in ICN

As stated in Section 1.2, the intermediate routers in ICN are allowed to cache the received data packets, thus are capable to satisfy future interests for the same content. This concept of ubiquitous caching introduced by ICN has enabled the study and proposal of multiple technologies which have shown their efficacy due to caching [125].

In general, the caching process of ICN can be divided into two phases: first, a placement algorithm that determines the selection of data packets which should be cached. Secondly, the replacement algorithm that takes the decision of evicting the data packets which are already in cache so that new data packets could be inserted when it is full.Some major placement algorithms are Leave Copy Everywhere (LCE), where all the data packets received by a router are stored into the cache, and Leave Copy Down (LCD) [122], where every time a router encounters a cache-hit, it execute the data packet in a way that it is cached one hop closer to the client. Another popular

approach for caching is ProbCache [166], where the caching decision of data packets depends on specific probability which directly relates to available caching resources along the path.

Forwarding and Matching Semantics in ICN

The forwarding phase in ICN is different from the forwarding phase in IP. It is depicted in Figure 1.1. In particular, upon receipt of interest for specific content, ICN router first checks whether the requested content is already present in the *cache* (i.e., Content Store). If the content is not found in the cache, the router looks in a *Pending Interest Table* (PIT) for a pending interest issued for the same content. Categorically, there are three possible outcomes in this case:

1. If the PIT entry for the similar name exists, and the arrival interface of the present interest is already in arrival-interfaces, the interest is discarded.

2. If the PIT entry for the similar name exists, yet the arrival interface is dissimilar, the router adds the new incoming interface to arrival-interfaces, and the interest is not forwarded further.

3. Otherwise, the router constructs a new PIT entry and forwards the interest utilising its FIB.

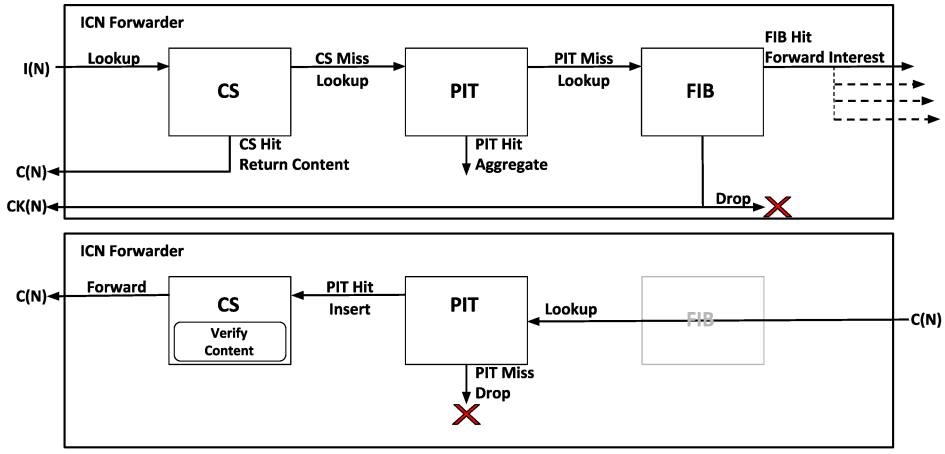

Figure 1.1: ICN interest and content forwarding logic

For the above described second outcome, when the requested content arrives at the router, all the pending interests for it are satisfied just by sending the content back to all the hosts who issued those interests. In

this way, ICN provides explicit support of multicast data routing, which indeed means huge benefit for receiver-driven multimedia delivery [128]. The router's *Forwarding Information Base* (FIB) is responsible for forwarding interests towards the content provider via one or more network interfaces (faces) based on the routes to the origin node(s). The requested data packet is then forwarded towards the sender by simply traversing, in reverse, the path of the preceding interest [222].

ICN Packet Format

As stated in Section 1.2, communication in ICN follows the request/response paradigm, i.e., request is the interest packet, while the response is the content packet. Both the packets are identified by the name of information. The validity of response lies in the fact that the content packets carry the same prefix name of the corresponding request, the interest packet. The routers use the name to forward the interest to the corresponding producer(s) and to forward the content back to requesting consumer(s). Figure 1.2 depicts the main fields of ICN packets, which are described below. A full list of fields can be found in [74].

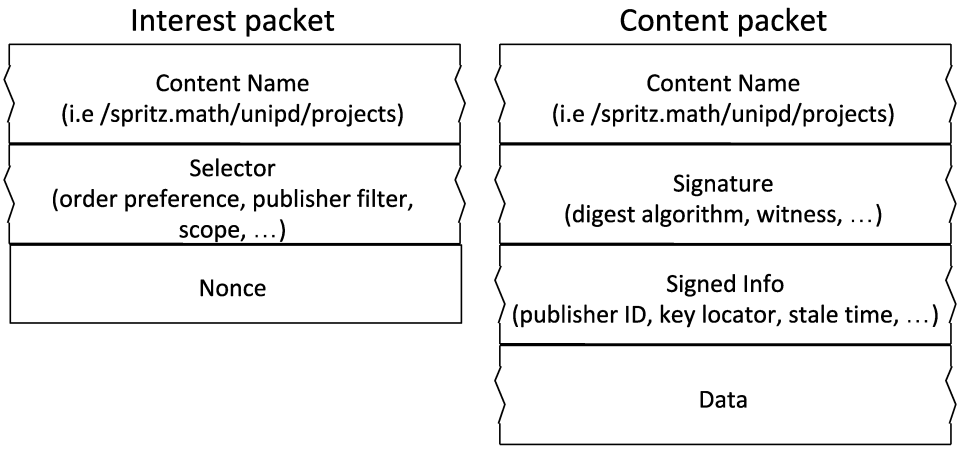

Figure 1.2: ICN packets

In the Interest packet, the name represents the hierarchical name for ICN content. Some more details about the name format which/some can optionally be adopted are given in the following:

- Selectors: it consist of `MinSuffixComponents/MaxSuffixComponents`, precisely, referring to the name of the name component. It allows the consumer to specify the whether the name is the full name including the digest, or the full name excluding the digest, or the content

name is known to be in a range of legitimate components. It also includes `PublisherPublicKeyLocator` which specifies the name of the key used to sign the corresponding data packet. `ExcludeFilter` can be used to choose whether to exclude list and/or ranges of name components from the responding content packet.

- Nonce: is a random number that uniquely identifies the Interest packet.

- Guiders: `Scope` and `InterestLifeTime` are used to limit the interest propagation and the time before which Interest expires.

Apart from the name of component which must be of the same corresponding Interest packet, the content packet is composed of the following fields:

- MetaInfo: contains the `ContentType` which could be by *default* specifying the actual data bits identified by the content name, and `LINK` which specifies the actual name of content. In addition, `KEY` is the public key required by producer.

- Content is the data itself. Signature is composed of `SignatureInfo` and `SignatureValue`. The first is included in the signature computation and describes the signature, signature algorithm, and other information such as the *KeyLocator*. The second is excluded from signature computation and is the actual bits of the signature and other supporting information. The signature is characterized by the `SignatureType` that could be *DigestSha256, SignatureSha256WithRsa, or SignatureSha256WithEcdsa,* and by the `keyLocator`. The latter is essential for retrieving the public key used to sign.

1.3 Proven Strengths of ICN

ICN has proved itself to be a potential candidate for a Future Internet Architecture. Efficient and scalable content distribution is one of the key advantages of ICN approach but is not the only which motivates the Internet to switch to a new infrastructure. In this section, we describe a subset of ICN advantages over existing technologies that reinforce motivation.

Scalable and Cost-Efficient Content Distribution

The mobile video preponderance is certain in future Internet traffic trends such as according to Cisco VNI estimation, video data will consume more than 80% of the IP traffic, and the wireless mobile devices will generate two third of the Internet traffic [209]. In fact, the video streaming services such as Netflix and YouTube together amount nearly 50% of Internet traffic. Similarly, other services such as Hulu, Amazon and HBO GO are gaining popularity as well. Due to this rapid increase in multimedia traffic over Internet, the network operators are facing challenges in meeting the bandwidth requirements of the end users. As a result, the networking paradigms which have been proposed to cache the traffic within the network aims to ease the bandwidth blockage. ICN is an emerging networking paradigm with an inherent support for caching at the network layer [23, 101].

The distinct features of ICN architecture such as receiver-driven mechanism, in-network caching, inherent support for mobility, and multi-cast routing makes it a perfect candidate to fit in the design space of receiver-driven multimedia streaming systems. Several recently proposed studies exposed the importance of ICN as a valuable alternative to TCP/IP in order to advance the competence of the current multimedia streaming systems [123, 125, 136, 164, 169, 179].

Mobility and Multihoming

ICN is an emerging networking paradigm, which meets 5G requirements such as global Internet access and seamless user mobility over dense and progressively heterogeneous network access by adapting to multiple radio access technologies, e.g., Wifi and LTE. ICN offers natural support for mobility at the network layer. It is due to the result of decoupling the time and space among request resolution and content transfer [76].

Two fundamental characteristics of ICN architecture encourages seamless consumer mobility [28, 76]. Firstly, the communication model is receiver(consumer)-driven instead of producer where the consumer uses the location-independent content names to request data. Secondly, the request/response communication model of ICN between consumer and producer is connection-less (i.e., *stateless*). Therefore, when a mobile consumer attaches to a new Point of Attachment (PoA), the above two characteristic permits consumer to re-issue interests to obtain the data, which it did not receive from its previous PoA. The producer mobility is instead more challenging in ICN because of no separation between routing locator and the content identifier. Several proposals for handling producer mobility exist in ICN literature [28, 223].

Disruption Tolerance

Achieving end-to-end communications with the transport sessions provided by TCP/IP is often difficult in challenged networks, i.e., networks having sparse connectivity, high-speed mobility, and disruptions. Since the application protocol sessions are bound to transport sessions, thus also fails as soon as the transport session fails. In existing Internet, several applications do not necessitate seamless communication with end-to-end paths [158]. As the primary objective is to access data objects, ICN is the perfect approach for Delay-Tolerant Networking (DTN) architectures [75, 194]. This is because ICN s in-network caching with hop-by-hop transport functionality provides a store-and-forward mechanism which provides better performance and reliability than using TCP/IP.

1.4 ICN Security

Learning from the past of TCP/IP, the current design of ICN architecture is putting more attention in including security from the outset. In the following, we describe the current state of the art of security in ICN. We present the content-based security model over which ICN has been designed so far as well as other architectural choices that carry some benefit in terms of security. We later address the security, privacy and access control issues that have not been solved yet in the current design.

Content Based Security

ICN is built on the notion of content-based security, i.e., protection and trust travel with the content itself, rather than being a property of the connections over which it travels. For this reason, every content packet is authenticated via digital signatures. The signature algorithm, used to sign content packets, is selected by the producer from a large scale of the fixed set and chosen to meet required performance in the signature/verification process. Every content packet moreover includes all the information necessary to verify the signature and retrieve the public key necessary. It can include cryptographic digest, or fingerprint of that public key; a shorthand identifier for the publisher; and a key locator to indicate where the key can be obtained, or containing the key itself. Regarding private content, ICN content packages are suitable for encryption techniques. In fact, it is possible to encrypt the payload of a content packet to protect content from unauthorized access. The well-known encryption techniques can be easily applied without requiring any modification.

Architectural choices Enhancing Security

Three architectural choices in ICN, namely PIT, CS and the pull-based communication model, can be considered a natural security enhancement for the ICN network layer. In particular, they are a simple countermeasure to bandwidth depletion attacks; a specific type of Denial of Service (DoS) attacks that exhausts the available bandwidth of a target victim thus hindering victims communications. In IP, bandwidth depletion is usually carried out from a set of bots controlled by an adversary. Each bot sends packets, either directly or through reflection attacks, at its maximum data-rate thus exhausting the victims network resources. The only way to counter this attack, without creating some damage to the genuine users, is to identify the source of the attack and block incoming packets from such source. A similar attack is hard to mount in a ICN networks [87]. First, the pull-base communication model prevents an attacker to forward content packets if the targeted consumer did not express an interest for them. Second, in case the attacker sends interest packets to the target victim, the interest collapsing mechanism and the in-network caching reduce the effectiveness of the attack. Closely-spaced interests requesting for the same content will be collapsed as soon as their paths cross the same router, and only the first of such interest will reach the victim. In case the adversary delays dispatching of interests requesting the same content to prevent their collapsing, the ubiquitous caching will copy the content in different parts of the network. This will prevent any interest crossing a copy of the content to reach the victim.

1.5 ICN Security Issues

ICN was designed in part to address several problems of IP-based networks. Content integrity, confidentiality, availability, access control, and privacy are all paramount to the success of ICN architecture. While ICN addresses some of these fundamental problems, it comes with its own challenges. In this section, we survey a subset of open problems.

- **Cache Poisoning:** One of the essential components of ICN is caching since the performance of the ICN foundation is based on receiver driven caching that strives to deliver the nearest available copy to the user. Thus, diminishes overall latency and increases bandwidth utilization for popular content. Nevertheless, despite its obvious benefits, content caching in routers opens the doors for the new type of attacks which are different from conventional DoS. For instance, the frequently requested

(popular) contents can be caches in the network to reduce request latency and network load. However, an attacker can undermine this popularity based caching by skewing content popularity and requesting less popular content more often. This specific attack is called a *cache pollution attack*.

Existing studies explores two types of cache pollution attacks: locality disruption and false locality [195]. In general, in the locality disruption attack, an attacker frequently requests new (unpopular) contents to disorder cache locality by churning the cache. On the other hand, in the false locality attack, the aim of the attacker is to change the popularity distribution of the local cache to favour a set of fake/unpopular contents. To do so, the attacker repeatedly requests the fake/unpopular contents. Categorically, the purpose of both attacks is to degrade cache effectiveness and increase content retrieval latency, e.g., decrease cache hit ratio, and consequently increase the average hop count of consumers requests.

Several proposed approaches in the literature aim to mitigate the attack. However, few of them incurs high computation cost at the intermediate routers, which undermines router's scalability [13, 162, 213]. Similarly, few proposed mechanisms either only detect the cache pollution attack [57] or address the less severe malicious provider attack scenario. Despite that previous studies have investigated the cache poisoning to aim a robust and efficient caching mechanism. Still, the interaction of cache and widely used existing applications/technologies (e.g., multimedia streaming, video-on-demand) needs immediate attention of the research community.

- **Cache Privacy:** Despite the great improvements that caching content packets brings on throughput, latency and network congestion; in-network caching can be abused to retrieve some information about the users. In particular, uncomplicated timing attacks can exploit ICN routers as "oracles" and allow the adversary to learn whether one of the nearby consumers recently requested certain content. If the adversary can then know who was the consumer that requested the content, it might be able to profile his activity. Similarly, probing attacks that target adjacent content producers can be used to discover whether certain content has been recently fetched [18].

- **Denial of Service:** Interest aggregation and content caching both provide limited protection against DoS. Nevertheless, this comes at a

price. For instance, the stateful forwarding via the PIT is also a major DoS vector. It is trivial for an active and malicious adversary to saturate the PIT with malicious or fake interests such attack is called *Interest Flooding Attack (IFA)* [49, 60, 87]. Since routers cannot differentiate between benign and malicious interests, they allocate space in their PIT to store pending interest information of malicious interests. Deterministically guarding ICN routers and network deployments against this type of attack is an open problem.

In ICN, another form of possible DoS attack is at the network's forwarding layer. Specifically, if the routing updates are not authenticated, the malicious adversaries may pretend ownership of the namespace(s) for which they are not entitled. Routers may also seize a namespace and block interests from reaching a legitimate producer. Defending against such attacks is an open problem in the absence of obligatory signature verification.

- **Access Control:** Access control is the mechanism through which only authorized consumers are given access to certain content. This is an application-layer problem since it involves consumer authentication. Due to in-network caching, access control security service is much more complicated in ICN. Generally, contents in ICN can be classified in two types: open access contents and restricted access contents. We are concerned about restricted access contents that should be accessed only by legitimate users. IP/TCP based access control mechanisms cannot be applied directly to ICN architectures because they need to recognise various fundamental features of ICN, e.g., native mobility support, in-network caching, and that ICN communication does not depend on IP addresses. Also, in ICN, access control mechanisms should address the various attributes of a "good" access control mechanism, e.g., reduce unnecessary communication overhead, should consider ICN-based mobile scenarios, minimize the exchange of secret keys, reduce extra operations for access control, prevent access control attacks and preserve the privacy of ICN users.

1.6 Contributions

In this thesis, we contribute to the security of ICN strengths. In particular, we address four different topics: vulnerabilities in the interaction of ICN's implicit features and widely used existing technology, such as multimedia streaming; how to utilize ICN's mobility support to provide security services

to the upper layer, such as authentication; securing the intrinsic mobility features offered by ICN; and architectural security issues that are intrinsic to the ICN design. In the last part of the thesis, we consider the feasibility and effectiveness of ICN with respect to real world deployment configurations. We provide analysis and classification of the solutions which aims at the coexistence of ICN and existing Internet architecture.

1.6.1 Secure and Efficient Adaptive Multimedia Streaming in ICN

The most adopted multimedia streaming method to enhance bandwidth utilization (e.g., adopted by Netflix, YouTube and HBO) is HTTP based Dynamic Adaptive Streaming (DASH). It provides a dynamic approach to time-shift control on media requests in response to fluctuating bandwidth conditions experienced by individual users. In particular, DASH strives to adopt the most appropriate resolution via real time bandwidth measurements in unstable network conditions, to deliver the best possible Quality of Experience (QoE).

Taking into account the significance of ICN in reducing bandwidth utilization, and to overwhelm the constraints of multimedia streaming, recently, the research community has investigated the implementation of Dynamic Adaptive Streaming (DAS) over ICN. Several studies have shown that ICN's receiver driven content delivery with in-network caching can significantly enhance the performance of adaptive multimedia streaming with DASH.

ICN in-network caching feature is beneficial for the content provider in terms of lower transmission delay and reduced bandwidth. However, we show that it makes Dynamic Adaptive Streaming (DAS) to be more challenging in ICN by exposing it to new security risks. Driven by the importance of addressing security issues in the initial stages of a potential new Internet architecture (i.e., ICN), we identify that an attacker can adversely exploit two fundamental ICN features, namely *in-network caching* and *interest aggregation*. In particular, the adversary is able to harm the adaptive behaviour of DASH streaming control system, which leads to the degradation of user perceived QoE. We believe that our proposed attack, which is supported by a comprehensive investigation is essential before ICN can be considered adequate for DAS, and it is deployed for real-world multimedia applications.

To address the above mentioned problem, we proposed two counter approaches. First, we propose a receiver driven approach called *Fair-RTT-DAS* to countermeasure the attack. Fair-RTT-DAS uses a scalable adaptive rate

control technique to enhance user perceived QoE in the presence of an adversary and ICN vital features. Fair-RTT-DAS maintains fairness[4] in the face of uneven round trip time (RTT) values caused by ICN dynamic in-network caching and implicit multicast support features. We show that utilizing Fair-RTT-DAS, the DASH client which is experiencing higher throughput variations due to varied content source locations or due to BOA, it will be able to switch with better resolutions leading to improved QoE.

In the second approach, we aim to eliminate the deficiencies of ICN architectural features related to caching and forwarding. We claim that for DAS, ICNs autonomous on-path cache management initiates enormous cache redundancy, results in sub-optimal selection of cached contents, and inherits networkwide cache-ignorant routing. Therefore, we propose to mitigate BOA based on timely and global knowledge of content access information. We implement Coordination with lightweight Monitoring for DAS to enable network-wide coordinated caching and cache-aware routing. By this, it aims to reduce bitrate oscillations and cache content redundancy in presence of both BOA and inherent content source variations, thus it enhances perceived QoE.

1.6.2 Authentication protocol for ICN based Mobile Networks

ICN is an emerging networking paradigm, which meets 5G requirements such as global Internet access and seamless user mobility over dense and progressively heterogeneous network access. Two important ICN design choices provides ICN a natural support for *consumer mobility*. First, content is addressed by location-independent human readable names, namely they do not express any reference to source or the destination of packets (both interest and content). Second, neither consumers nor producers require a network address (e.g., the IP address) to communicate. Only the name of content is used to forward consumer's interests towards the corresponding content, and the content back to the requesting consumer. This allows consumers to forward interests as soon as an interface is available, as opposed to IP in which a host is forced to wait for a mapping between the interface address and its layer-3 address. Such content-based, location-independent communication style has been shown to improve device mobility support

[4]Unlike TCP/IP where fairness implies equal resource allocation to multiple flows, we use the term *fairness* to depict optimization of throughput in a single flow by smoothing the RTT of received data packets.

with respect to the current IP [170], thus raising ICN as a possible future solution to manage mobility at network layer.

In this work, we propose a simplified LTE infrastructure that exploits the ICN architecture to manage device mobility. Inspired by recent proposals that manages mobility at ICN network layer [32, 224], we present a simplified LTE architecture that does not require the Mobility Management Entity (MME); i.e., the entity that guarantees an uninterrupted device connection during mobility events. We use the ICN communication style to design a revised device authentication protocol and a novel handover authentication protocol that reduces the number of exchanged messages between the authenticating entities. In particular, our handover authentication protocol exploits the ICN synchronization protocol [227] to move the device security context (i.e., cryptographic material established during the mobile device authentication) during the handover mechanism from the old to the new base station. Our analytical evaluation shows that our authentication and handover authentication protocols reduces the device authentication delay when compared with the current LTE authentication protocols.

1.6.3 Secure Mobility Management in ICN

The *producer mobility* is instead more challenging in ICN because of no separation between routing locator and content identifier. Several proposals for handling producer mobility exist in ICN literature [28, 223]. Among them the *routing(tracing)-based* approaches try to address the subject by updating the forwarding tables at each mobility event, and then forwards the request. Employing the routing-based protocols [32, 95, 109, 206, 224], the producer is entitle to directly exploit ICN stateful forwarding plane to provide seamless mobility and to overcome the handoff latency, packet loss, and signalling overhead. Since tracing-based protocols allow producer to directly interact with the network forwarding information, therefore, installing such protocols which are deprived of acceptable security mechanisms bring up serious security threats for all network entities, i.e., consumer, producer, and the network itself. In this regard, the producer should be allowed to issue only the legitimate routing updates, explicitly named as *Interest Updates (IUs)*, for the prefix(es) that it is entitled to publish the relevant content. In cases where no adequate security mechanism exists to impose such rules then an adversary is able to easily forge IUs of the legitimate producers. Hence, it can divert benign consumers requests and network traffic towards itself, such attack in ICN is known as *prefix hijacking* [33]. By launching prefix hijacking, an adversary is able to: (i) victimize benign users by per-

forming blackhole attack [25], (ii) deny consumer's access to their requested content [89], (iii) make genuine content reachability unavailable, and (iv) pollute the network caches with false content [58].

In this work, to address the above-mentioned security threats, we propose a Blockchain based lightweight distributed mobile producer Authentication (BlockA*uth*) protocol to enable secure and efficient mobility management in ICN. BlockA*uth* authenticates the producer prefix(es) and enforce them to express only genuine routing updates for the prefix(es) which they are entitled to advertise. Our qualitative security analysis confirms that BlockA*uth* is robust against various security attacks to which mobile network and blockchain are particularly vulnerable (e.g., prefix hijacking, double spending, DoS attack). In addition, the performance evaluation of BlockA*uth* shows that it maintains significant performance gain compared to the state-of-the-art prefix attestation proposals. In particular, it maintains up to 94% of the network's original throughput, while just requires additional storage of tens of megabyte.

1.6.4 Denial of Service Mitigation in ICN

One of the key goal of ICN is "security by design", however, attackers have exploited the ICN design features to build novel attacks and introduced a new type of Distributed Denial of Service (DDoS) attack, better known as Interest Flooding Attack (IFA). In IFA, an adversary issues non-satisfiable requests in the network to saturate the Pending Interest Table(s) (PIT) of ICN routers and prevent them from properly handling the legitimate traffic. Several detection and reaction mechanisms try to mitigate this problem [19, 49, 60, 87, 172], but all of them are not highly effective and, on the contrary, heavily damage the legitimate traffic.

In this work, we propose a novel mechanism for IFA detection and mitigation, aimed at reducing the memory consumption of the PIT by effectively reducing the malicious traffic that passes through each ICN router. In particular, our protocol exploits an effective management strategy on the PIT which differentially penalizes the malicious traffic by dropping both the inbound and already stored malicious traffic from the PIT. To evaluate the effectiveness of work, we implemented our proposed protocol on the open-source ndnSIM simulator and compared its effectiveness with the one achieved by the existing state-of-the-art. The results show that the proposed protocol effectively reduces the IFA damages, especially on the legitimate traffic, with improvements that go from 5% till 40% with respect to the existing state-of-the-art.

1.6.5 A Survey on ICN-IP Coexistence Solutions

The benefits of Information Centric Networking can occur only in a full-ICN scenario, which implies a complete replacement of the current Internet. Despite its obvious need, this is a long and complex process, that requires the coordination among the different parties (ISPs) on various attributes, e.g., time, costs for updating hardware and software of the network components and ability to face all the new possible challenges. Previous attempts to replace a widely used technology, protocol or architecture have always faced a long period of coexistence between the old and the new solution. In the same way, the replacement of the current Internet will involve a *transition phase* during which IP and ICN architectures will coexist. Researchers working in this field have already addressed the coexistence of ICN and IP following various techniques. However, to design a complete coexistence architecture, it is first necessary to have a comprehensive overview of the strengthens and weaknesses of the existing solutions.

In this work, we provide a comprehensive analysis and classification of the existing coexistence solutions. In doing so, we define a set of relevant features and evaluation criteria which are necessary to analyze a coexistence architecture. We also provide a comprehensive analysis and comparison of all the main coexistence solutions. Lastly, we discuss the open issues and challenges affecting the existing coexistence architectures and provide some possible insights to overcome them.

1.7 Thesis roadmap

The organization of this dissertation is as follows. In Chapter 2, we present the novel vulnerability in adaptive multimedia streaming and proposed countermeasures for it. The brief background information about dynamic adaptive streaming and related work with respect to ICN is discussed in Section 2.1. Section 2.2 describes the system and adversary models for this work. The proposed attack and respective receiver-driven mitigation technique have been presented in Sections 2.3 and 2.4, respectively. The result evaluation and analysis of proposed attack and receiver-driven mitigation are presented in Section 2.4.4. In Section 2.5, we present the architectural frame work for efficient and robust dynamic adaptive streaming in ICN. Section 2.5.1 describes the working methodology of proposed network based mitigation along with functional details. The evaluation and effectiveness of proposed architecture are detailed in Section 2.5.4. Finally, we conclude the chapter in Section 2.6.

In Chapter 3, we present the efficient device authentication protocols for cellular network exploiting ICN's native mobility features. In Section 3.1, we present the LTE authentication protocol and the handover mechanism. Then, in Section 3.2, we present proposed simplified LTE infrastructure onto which we designed the proposed authentication protocols along with their functional details. In Section 3.3, we evaluate our proposed authentication protocols comparing to existing LTE authentication protocols. In Section 3.4, we provide a security discussion of our protocol. Finally, we conclude this chapter in Section 3.5.

In Chapter 4, we focus on the fundamental security issues in ICN's mobility management and provide BlockChain based secure and fast mobility framework in ICN. Section 4.1 presents the mobility-enabling technologies in ICN. Section 4.2 discusses the fundamental security challenges with respect to ICN mobility management. Section 4.3 provides a brief overview of BlockChain technology. Section 4.4 describes the design and working methodology of our proposed secure framework for mobility management, i.e., BlockA*uth* protocol, and Section 4.5 illustrates a scalable BlockChain solution for BlockA*uth*. Sections 4.6 and 4.7 describe the security and performance analysis of BlockA*uth*. Finally, we conclude in Section 4.8.

In Chapter 5, we focus on specific DoS attacks in ICN, i.e., Interest Flooding Attack (IF) and propose a countermeasure to it. In Section 5.1, we discuss the IFA. In Section 5.2, we describe the existing solutions for IFA mitigation in ICN. Section 5.3 briefly describes the proposed mitigation protocol including system, adversary model and working methodology. In Section 5.4, we present the evaluation and comparison of ChoKIFA against IFA and state of the art. Finally, Section 5.5 presents the summary of the work.

In Chapter 6, we present an analysis on the deployment configurations of ICN architectures in coexistence with existing Internet. In Section 6.1, we illustrate the basic concepts of the TCP/IP protocol suite in comparison with the ICN. Section 6.2 describes the criteria which we identified and used for the analysis and classification of the coexistence architectures. In Section 6.3, we illustrate each coexistence architecture and provide the motivation for our classification. In Section 6.4, we discuss the main strengths and limitations of the current coexistence architectures, and provides insights for improving the design of the transition phase. Finally, we conclude the chapter in Section 6.5.

Finally, Chapter 7 concludes this dissertation and we propose some future direction in the topic of Securing Information Centric Networking.

1.8 List of publications

Part of the research presented in this thesis and developed during my PhD program produced peer-reviewed conference and journal publications. In this section, we report the complete list of published and currently submitted works. Publications are listed in chronological order.

Conference and Workshop Publications

[C1] Alberto Compagno, Mauro Conti, Muhammad Hassan Khan. An ICN-based Authentication Protocol for a Simplified LTE Architecture. *In proceedings of WCS Eurocrypt workshop*, France, April 30, 2017.

[C2] Mauro Conti, Ralph Droms, Muhammad Hassan, Sebastiano Valle. QoE Degradation Attack in Dynamic Adaptive Streaming over ICN. *In Proceedings of the 19th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (IEEE WoWMoM 2018)*, Chania, Greece, June 12-15, 2018. (GGS: 3/B; CORE: A; LiveSHINE: C; MA: B)

[C3] Muhammad Hassan, Hani Salah, Mauro Conti, Frank H. P. Fitzek, Thorsten Strufe, CoMon-DAS: A Framework for Efficient and Robust Dynamic Adaptive Streaming over NDN. *In Proceedings of the 24th IEEE Symposium on Computers and Communications. (IEEE ISCC 2019)*, Barcelona, Spain, June 29 - July 3 2019. (GGS: 3/B; CORE: B; LiveSHINE: A-; MA: B)

[C4] Abdelmadjid Benarfa, Muhammad Hassan, Alberto Compagno, Eleonora Losiouk, Mohamed bachir Yagoubi, Mauro Conti. Choose to Kill IFA (ChoKIFA): A New Detection and Mitigation Approach against Interest Flooding Attacks in NDN. *In Proceedings of the 17th International Conference on Wired/Wireless Internet Communications. (IFIP WWIC 2019)*, Bologna, Italy, June 17-18 2019. (LiveSHINE: C; MA: C; ERA: B)

Journal Publications

[J1] Mauro Conti, Ralph Droms, Muhammad Hassan, Chhagan Lal. Fair-RTT-DAS: A robust and efficient dynamic adaptive streaming over ICN. *In (Elsevier) Computer Communications*, 129: 209-225, 2018. DOI: 10.1016/j.comcom.2018.07.033, ISSN: 0140-3664. (JCR IF 2017: 2.613; IT-ANVUR Class: 2)

[J2] Mauro Conti, Muhammad Hassan, Chaggan Lal. BlockAuth: BlockChain based Distributed Producer Authentication in ICN. *In (Elsevier) Computer Communications, 2018.* (Under submission.)

[J3] Mauro Conti, Ankit Gangwal, Muhammad Hassan, Chhagan Lal, Eleonora Losiouk. The Road Ahead for Networking: A Survey on ICN-IP Coexistence Solutions. *IEEE Communications Surveys and Tutorials (COMST), 2019.* (Under submission.)

Chapter 2

Secure and Efficient Adaptive Multimedia Streaming in ICN

To sustain the adequate bandwidth demands over rapidly growing multimedia traffic and considering the effectiveness of ICN, recently, HTTP based Dynamic Adaptive Streaming (DASH) has been introduced over ICN, which significantly increases the network bandwidth utilisation. ICN in-network caching feature is beneficial for the content provider in terms of lower transmission delay and reduced bandwidth. However, we show that it makes Dynamic Adaptive Streaming (DAS) to be more challenging in ICN by exposing it to new security risks. Driven by the importance of addressing security issues at the initial stages of a potential new Internet architecture (i.e., ICN), we identify that an attacker can adversely exploit two fundamental ICN features, namely *in-network caching* and *interest aggregation*. In particular, the adversary is able to harm the adaptive behaviour of DASH streaming control system, which leads to the degradation of user perceived QoE.

In this chapter, we investigate on the novel vulnerabilities in ICN based Dynamic Adaptive Streaming (DAS). Then, we proposed two countermeasures to mitigate the vulnerability and provide efficient and robust DAS: (i) a scalable receiver-driven approach and, (ii) an efficient network architecture which eliminates the demerits of ICN features for multimedia streaming. We believe that our work, which is supported by a comprehensive investigation is essential before ICN can be considered adequate for DAS and it is deployed for real-world multimedia applications.

2.1 Dynamic Adaptive Streaming

In recent years, DAS has become the most used adaptive bitrate streaming technique that supports on-demand and real-time multimedia streaming. Therefore, most of the Internet video streaming providers such as Netflix, Amazon and Sky rely on DAS [169]. Along with solutions like Apple HTTP Live Streaming, Microsoft Smooth Streaming, and Adobes HTTP Dynamic Streaming which were highly appreciated, MPEG-DASH (Dynamic Adaptive Streaming over HTTP) is ratified by ISO/IEC and it became the utmost used standard for DAS. DASH specifies the description of multimedia content availability and the process of how it shall be segmented.

Figure 2.1 shows the idea of adaptive multimedia streaming over HTTP. The figure indicates that media content is encoded in different versions, and it provides variability in bitrates, resolutions, codecs, and so on. These versions are further sliced into segments of specific lengths, and the client adopts a pull-based approach to request each segment individually using HTTP GET requests [125]. In particular, the multimedia content on the HTTP server entails two distinct elements, namely: segments and Media Presentation Description (MPD). The relationship between a segment's associated characteristics (e.g., bitrate, resolution, codec, timeline) and the location are provided by the so-called XML based MPD, where HTTP URL represents an individual segment. Initially, the MPD file is retrieved by DASH client. Thus, using the information in MPD file, the DASH client requests the most appropriate bitrate by considering the user's current context, i.e., bandwidth fluctuations, preferences, etc. [128]. As a consequence, the streaming system is pull-based, and the entire streaming logic is located on the client, which makes it scalable, and possible to adapt the media stream to the client's capabilities.

This dynamic adaptation of bitrate is based on different DAS strategies, which are mainly classified in two immense families: (i) Rate Based (RB), and (ii) Buffer Based (BB), referring that bitrate adaption processes either by estimating the throughput level or the buffer level. There are several representatives of these strategies. For instance, Probe and Adapt (PANDA) [131] and Buffer Occupancy based Lyapunov Algorithm (BOLA) [187, 188] are very popular, and mostly characterized as the benchmark for RB and BB in the literature. Despite this generalized categorization of DAS strategies, both metrics, i.e., throughput and buffer level, are often collaboratively used together in order to attain an improved adaptation process, named as Rate and Buffer based (R&B).

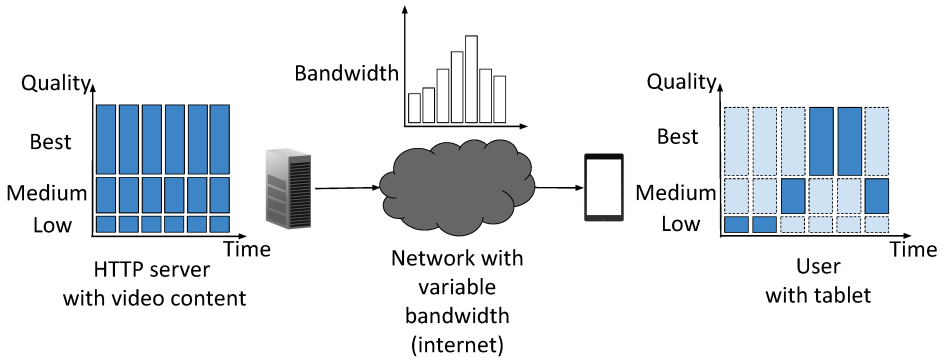

Figure 2.1: Dynamic adaptive multimedia streaming principle

2.1.1 Dynamic Adaptive Streaming over ICN

With the rise of ICN [101, 222] as a new communication paradigm for the future Internet and the popularity of adaptive multimedia streaming over HTTP [128], DASH over ICN has gained significant attention from the research community. Numerous works [123, 125, 136, 164, 169, 179] have considered in-network functionalities offered by ICN as a support for DAS. For instance, the authors in [64] shows an integration of DASH and ICN by enabling a proxy service between HTTP and ICN. Authors in [125, 136] fully exploit the potential of ICN and shows the implementation of DASH client as a native ICN interface. In particular, the framework transforms the HTTP request and reply messages to corresponding interest and content messages. Figure 2.2 presents the proposed architecture of DASH over ICN [101], where DASH-related components are marked in light blue and ICN-related components in dark grey colour.

In general, similar to HTTP-based multimedia streaming, the MPD defines the relationship between a segment's associated characteristics and its name. As used in DASH over ICN, each MPD lists the ICN names (i.e., Uniform Resource Identifier) of the media segments instead of URLs [124]. The ICN naming scheme in DASH over ICN supports versioning and segmentation, which is necessary for multimedia streaming in ICN [125]. The versioning of segments in ICN indicate different representations of DASH-based multimedia content. Figure 2.3 illustrates the example of CCN/NDN versioning, which is chosen to map different representations of DASH segments [125]. For instance, the *representation* 1 and 2 are directed to the versions of CCN URI denoted by $\_v1$ and $\_v2$, respectively. Similarly, DASH segmentation structure for the content is also supported by the ICN naming

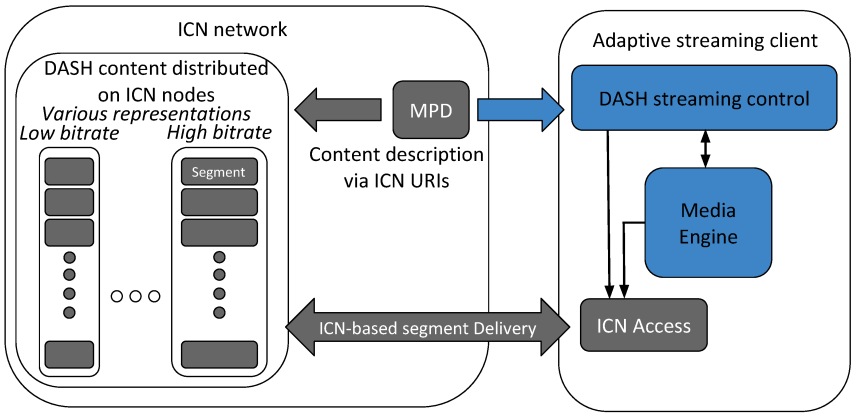

Figure 2.2: Dynamic adaptive multimedia streaming over ICN

scheme. Hence, a DASH client can request the content appropriate to the estimation of the available bandwidth and network conditions [128]. ICN interest messages are issued to retrieve video segments, and in return, the video segments are provided either by the original content source or returned from in-network caches of routers.

The role of the DASH streaming control mechanism is to adapt the client requests based on the available bitrate and network bandwidth. Therefore, it provides a smooth streaming session with high Quality of Experience (QoE). The work in [123, 125] shows that DASH over ICN is able to compete with the existing HTTP streaming system in terms of average download bitrate. Also, it is able to provide smooth streaming with reduced bandwidth requirements for the origin server. Furthermore, the authors in [136] demonstrate the usefulness of in-network caching in the presence of multiple clients fetching the same content, and resulting in increased video quality over time. Furthermore, the implementation of ICN-based dynamic adaptive streaming exhibits advantage while using Scalable Video Coding (SVC) [164], showing that layered approach increases the efficacy of adaptation process and guarantees a smooth playback without stalling.

We observe the behaviour of DASH streaming control system while interacting with ICN implicit characteristics. The effective support of in-network caching for popular content and native multicast capability in ICN greatly reduces the traffic burden for the content providers. However, these features also create new opportunities for the malicious users to degrade the QoE of a DASH client during its streaming process.

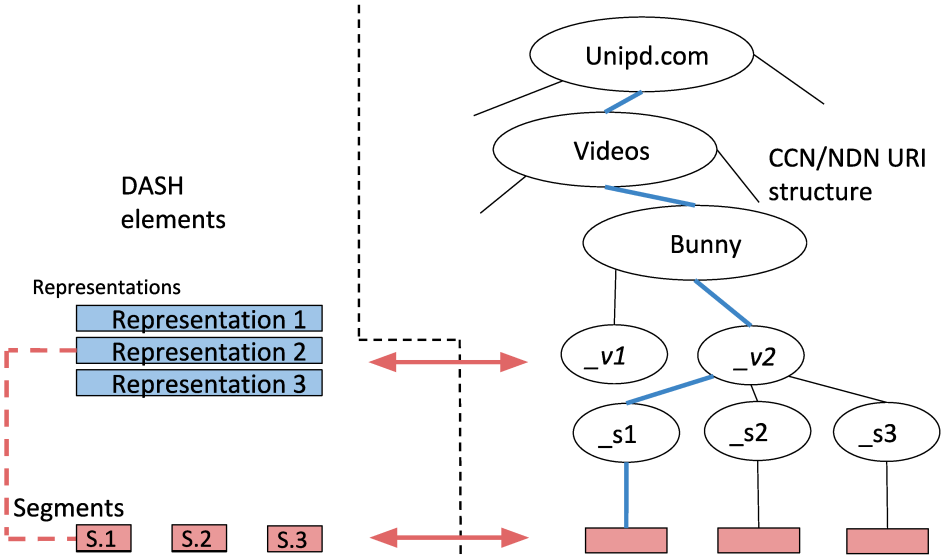

Figure 2.3: DASH's representation mapping to the versions of ICN's naming scheme

2.2 System and Adversary Models

In this chapter, we consider the scenario of multimedia streaming in ICN as illustrated in Figure 2.4. The content provider server called *producer* (P) stores the multimedia data (S) in a DASH-compatible format. S contains a collection of n number of equal-length segments, where each segment is available to be streamed in several media encoded bitrates (b), i.e., resolutions. A client (C) request the segment(s) from P in one of the available bitrates of S.

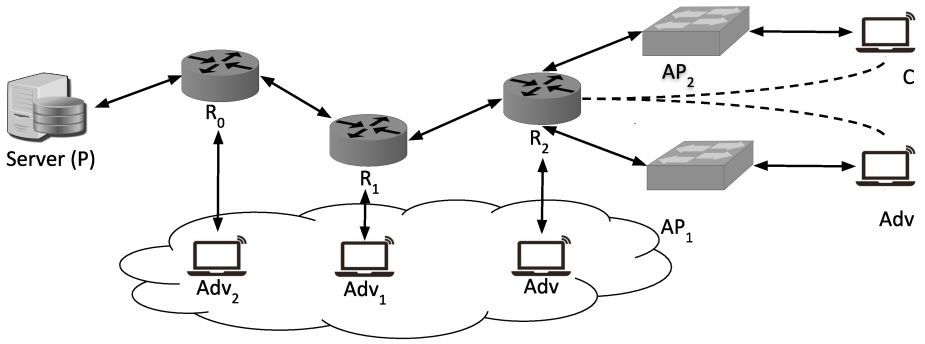

Figure 2.4: Topology considered

In our scenario, the adversary (Adv) is aware of the media content(s) being requested by C in advance. Each interest from C and Adv traverses one or more routers before being satisfied by P or one of the ICN router, R. The goal of Adv is to degrade the QoE perceived by C during the streaming process. In our scenario, every router operates according to the default settings of ICN [222]. Moreover, the forwarding strategy adopted is *bestRoute* [169], which routes packets with respect to lowest path cost. Summary of the notations used in our explanation are mentioned in Table 2.1.

Table 2.1: Summary of notations used

Notation	Meaning
Adv	adversary
C	client
P	server providing video
R_l	routers
AP	access point
S	video file at P
N	total number of segments in video
S_n	n^{th} segment of S
CS_l	cache at R_l
$b_{(i,j)}$	set of available bit rates of S
α	consecutive gap of variable length
b_f	bitrate of S received from CS
b_j	maximum bitrate
b_i	minimum bitrate
MPD	media presentation description (XML file)
$x(t)$	interest packet sending rate
$w(t)$	client interest sending window
c	link capacity
$d(t)$	inter-frame interval
T	sliding window time interval
s	MTU/fragment size
x	pre-defined constant value

2.2.1 System Model

In our scenario, all the entities, C, Adv, P, and R implement the ICN stack. For adaptive multimedia streaming in ICN, C and P use the aforementioned DASH over ICN model [125, 136]. Adv is aware of DASH but will make requests for video content to degrade the QoE of the content streamed by

C, rather than trying to optimize its own content delivery. We use two types of coding formats for DASH-compliant multimedia content: Advanced Video Coding (AVC) [126] and Scalable Video Coding (SVC) [115]. In AVC, each segment of different bitrates is represented by a unique segment name. For instance, the first segment of the 100 kb/s representation is referenced as `/dash/bunny/_2s_100kbit/bunny_2s1.m4s`. In SVC, video content is encoded in different independent layers of quality called as the base layer (BL) and the enhancement layers (EL), where each layer subsequently enhances the video quality. The segments are referenced by the MPD file, in which these segments are listed by their URIs [124]. C requests the segments according to the DAS streaming control system, which implements the adaptive performance. We investigate the behaviour of C in the presence of an Adv while using all type of adaptation strategies that are referenced as standard in DASH streaming control system, i.e., Rate-Based (RB), Buffer-Based (BB), and Rate-Buffer-based ($R\&B$) [116]. Below we briefly describe the functionality of these DAS adaptation logic techniques.

Rate-Based adaptation logic: The RB adaptation algorithms [102, 131] stand on the idea of using the previous segment's measured bandwidth as a measure of bandwidth estimation for next segment. This is because C measures the available bandwidth on each instance while downloading the segment. By using an exponential moving average, C can estimate the available bandwidth for the next segment using the Equation 2.1.

$$\lambda_{k+1} = (1 - \beta) * \lambda_k + \beta * \lambda, \tag{2.1}$$

where λ_{k+1} denote the new estimate for bandwidth and λ_k denote the previous estimate. λ denote current bandwidth, which is calculated by taking the ratio of current segment size to its download time. β is a constant which reduces the impact of fresh measures on the estimate. This information supports C to select the highest affordable media encoded bitrate b_{k+1} as a requesting bitrate, i.e., $b_{k+1} < \lambda_{k+1}$.

Buffer based adaptation logic: The BB adaptation logic function is independent to bandwidth estimation instead it selects the video quality according to the current buffer occupancy $B(t)$. The buffer is divided into multiple levels and C requests the b_{k+1} according to its actual buffer level. We use *Bandwidth independent Efficient Buffering* (BiEB) [187] as a standard in our model with a maximum buffer limit of 33 seconds.

Rate and Buffer-Based Adaptation logic: The R&B adaptation algorithm [24] proves to be a stronger coexistence between RB and BB decision

techniques. In this algorithm along with the bandwidth estimation for next segment λ_{k+1}, C also goals to stabilize the buffer level $B(t)$ around a target value (B_{max}). This keeps the adaptation process as smooth as possible by avoiding to react on short-term bandwidth spikes and stalling. In particular, the algorithm functions use two threshold values (B_{min} and B_{max}) along with λ_k and $\lambda_k + 1$. The increase/decrease of the video quality is governed in two ways. Decrease when $B(t) > B_{max}$ then algorithm keeps the current b. When the buffer lever is $B_{min} \leqslant B(t) \leqslant B_{max}$, it quickly shifts to a lower quality. Furthermore, the lowest quality is requested when $B(t) < B_{min}$. Conversely, if the buffer level is $B_{min} \leqslant B(t) \leqslant B_{max}$ or greater than B_{max} the quality is increased with respect to estimated bandwidth.

2.2.2 Adversary model

In our analysis, we assume *Adv* connects to the same first-hop router to which C is connected or to some on-path router(s) between P and C. By using geo-locating techniques [50], we could even relax the first assumption and require only that *Adv* just connects to the closest router. Using these existing techniques [50], *Adv* can identify the router closest to the consumer. In Section 2.4.4.3, we show the impact on C when the *Adv* launches the attack by connecting to different locations in the network. However, the maximum adverse impact to the victim results when *Adv* is connected to the same first-hop router, which later we take trademark in the rest of chapter.

We assume that *Adv* has prior knowledge of the multimedia content (S) that C will be requesting in near future. Several existing techniques support this assumption apart from the preliminary knowledge required to execute the attack subjective to C. *Adv* can exploit timing attacks as a side channel to breach privacy and infer if that content has been previously requested by C [18]. Moreover, *Adv* could also probe the MPD by exploiting the timing attacks to discover whether C has previously requested it or not. These techniques allow *Adv* to predict the video that C is going to request. Furthermore, it is quite possible that C may share the same wireless link, so the traffic traces are exposed and may be easily eavesdropped [219]. The *Adv* could infer the online activities of a user by analyzing the traffic and then it can be able to predict the content and its source [132].

2.3 Proposed Bitrate Oscillation Attack for DAS over ICN

In this section, we present details of our proposed attack, which degrades the user perceived QoE while streaming multimedia content in DASH over ICN. In order to degrade the QoE for C during the streaming session, Adv creates oscillations in the adaptive behaviour of the DASH streaming control system. The Adv implements the attack by forcing the DASH control to dynamically switch between high and low representations frequently (e.g., b^{++} and b^{--}), as shown in Figure 2.5.

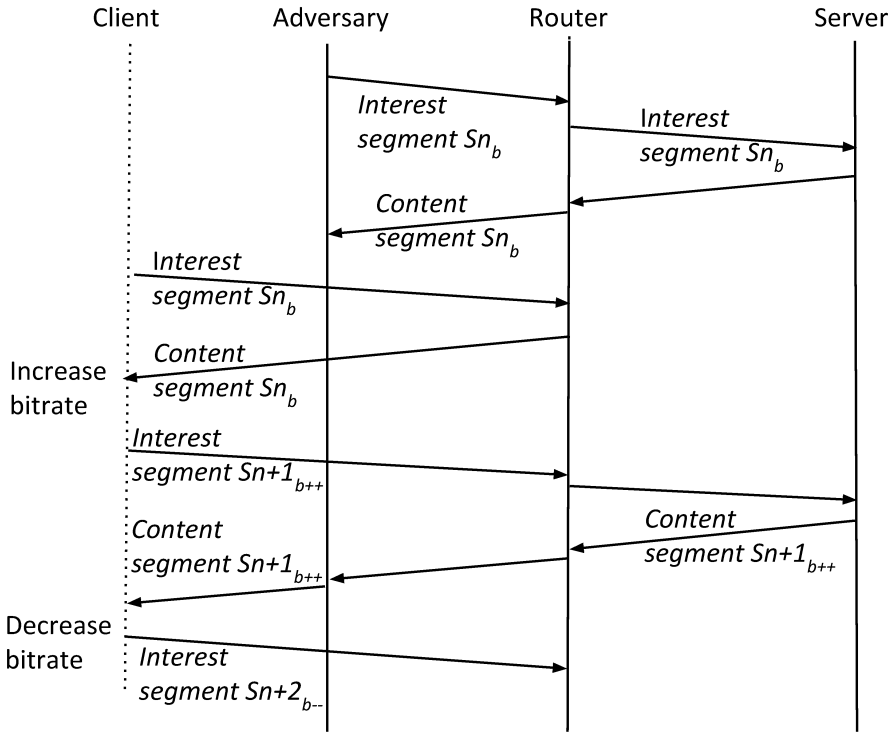

Figure 2.5: Interest sequence to perform bitrate oscillation attack

In summary, suppose C wants to stream a video file, S. To trigger oscillations, Adv requests selected segments of S in advance of requests from C. The segments returned from P through the network to Adv are stored on each intermediate router that earlier forwarded the corresponding interest [101]. When C subsequently requests the segments of S in sequence, some of those segments are returned from P and some from the intermediate routers. The difference in the times required to deliver the segments to C will

cause oscillation in the quality of the segments requested by C, degrading the QoE of the delivered video stream.

We assume that Adv has knowledge of the video stream S that will be requested by C. However, Adv is not aware of the specific bitrates of the segments of S that C will request. Initially, Adv receives the MPD containing the list of the all available segments of S in their various encoded representations (bitrates) [136]. As Adv implements its attack, it uses the list of segments in MPD to request a non-sequential subset of S in such a way which leads to higher oscillations for C. For each selected segment of S, Adv requests all available representations of that segment to ensure that any version of that segment requested by C will be cached in an intermediate router. Recall that, as the interest traverses through each router which is on the path from Adv to P, the routers create a state in the form of Pending Interest Table (PIT) entry to satisfy the requirements of interest aggregation [222]. After receiving the interest, P injects the requested content back into the network, which follows the same route from which the interest is received.

To perform the attack, Adv issues legitimate interests for S in a specific order. Each interest requested by Adv is for a new segment, and it requests all the available bitrates (b) of that segment. The interest aggregation on ICN routers (specifically, in NDN/CCN) always aggregate similar interests and forward a single request on their behalf. In our attack scenario, since each interest requests are for a new S_n, therefore, each S_n is always counted as a new entry in the PIT, and thus it is forwarded to the upstream routers. Adv exploits interest aggregation to store all the legitimate interests on all the on-path router's PIT until the content is received back for those respective interests. In this way, by issuing different legitimate interests in the PIT of all on-path routers for each S_n (in all available bitrates, $b_{(i,j)}$), Adv can store segments of relevant interest on intermediate routers cache.

In our model, Adv requests the segments of S in ascending order, skipping a number of segments between each requested segment. As shown in Figure 2.5, when C subsequently requests a segment of S, either Adv has previously requested that segment or the segment was one of the segments skipped by Adv and, therefore, not requested. Consider a specific segment, S_n, requested by C. If Adv has previously requested S_n and it has been returned from P, a copy of S_n will be available in the cache of R_2 and will be returned to C, in the round-trip time between C and R_2. If Adv has previously requested S_n but the segment has not yet arrived at R_2, the interest from C will be aggregated at R_2 and S_n will be returned to C in less than the full round-trip time between C and P, depending on where S_n is

on the path from P to Adv. If S_n has not previously been requested by Adv, the request will be forwarded to P and S_n will be delivered in the complete round trip time between C and P.

When subject to this attack, C interprets the relatively short delivery time for segments pre-fetched by Adv, as indicating high available bandwidth in the network. In contrast, while C interprets the longer delivery times for other segments, as indicating low available bandwidth. By pre-fetching segments of S with the consecutive gaps, Adv causes the DASH adaptation strategy at C to frequently switch between different bitrates, such as very low and high.

To better illustrate the attack and its corresponding behavior at C, let's consider the topology in Figure 2.4 where two DAS-enabled clients Adv and C are connected to R_2 via AP_1 and AP_2. The router R_2 has a cache, say CS_2, that can hold the multimedia content S originating from P. S is available in single-layer coding formats such as Advanced Video Coding (AVC) and P provides multiple representations of each segment, here each representation features to a set of resolutions ($Sn_{(b_{i\cdots j})}$). These representations are listed in the MPD of S and are organized with respect to the ICN naming scheme for the segments [125]. Following the adaptive streaming control, C request a representation that best matches with its current network conditions, and it can adapt to fluctuations in the network bandwidth by switching to lower or higher representations. The Adv and C are streaming the content S from P. The Adv launches BOA at time t_i, and the C starts streaming at time t_j, where $t_j > t_i$. We now present the design and configuration of our BOA Algorithm 1 running at Adv. It requests content in an ascending order with a predefined consecutive gap (α). We then investigate the behavior of C in the presence of our algorithm running at Adv.

After retrieving the MPD at time t_i, Adv issues a series of interests in ascending order, i.e., $S(n+\alpha)$, where α is the consecutive gap of variable or fixed length to issue the discontinuous requests toward P with all the available bitrates ($b_{(i,j)}$) of S. Each request issued by the Adv is for a new segment with all available bitrates ($b_{(i,j)}$) till it requests a total of N number of segments. For each interest, say $S(n+\alpha)_{b_{(i,j)}}$, routers first check if the corresponding segment is available in their cache. Than check if the interest is listed in the PIT after that router forward it to FIB [101]. This is because the interest is not previously requested, and there is no segment relevant to it in CS. Therefore, each router lists the entry in PIT for these interests and forwards them to P. These interests traverse the path $Adv \rightarrow Ap_1 \rightarrow R_2 \rightarrow R_1 \rightarrow R_0$. Following the characteristics of ICN [101], the segments in response to the interests follow the reverse path and these are

Algorithm 1 Algorithm for Adversary (Adv)

procedure SEQUENCE_OF_INTEREST (S, i, j, α)
  $MPD \leftarrow Send\_requests\_to\_P$ ▷ $MPD = \{S(n)_{b_{i,j}}\}$
  for $n = 1, n \leqslant N, n + \alpha$ **do**
    $Content(S(n)_b) \leftarrow Interest(S(n)_b)$
    for $i \leqslant f \leqslant j$ **do**
      $Content(S(n)_{b_f}) \leftarrow Interest(S(n)_{b_f})$
      ▷ $S(n)_{b_i}, \ldots S(n)_{b_j}$ *caches on* CS_k
    end for
    end for
  end for
  end for
end procedure
close;

cached at intermediate routers. At time t_j, C retrieves the MPD for the desired content. Based on the MPD, C sends interest messages to fetch S_n with an appropriate bitrate (say b_k) which is estimated based on its current networking conditions [129]. During the streaming session, when C request an interest $S(n + \alpha)$, the path that this specific interest will traverse is $C \rightarrow Ap_2 \rightarrow R_2$. It is because R_2 will return the content from its cache (CS_2). Let B_f be the bitrate representation of the segment received from CS_2, and b_{f+1} be the next, and b_i, b_j be the available lowest and maximum bitrate. For the interest $S(n + \alpha)$, C finds the download rate higher than the previous segment. Due to this, the DAS adaptation logic at C believes that the available bandwidth is suitable for receiving the maximum representation for next segment, and it switches to a highest representation, i.e., b_j. Now C requests $S(n+\alpha+1)$ with the maximum bitrate b_j, however, the interest explicitly traverses toward P through the path $C \rightarrow Ap_1 \rightarrow R_2 \rightarrow R_1 \rightarrow R_0$. This is because the requested segment has not yet been cached at R_2. Consequently, upon reception of $S(n+\alpha+1)$ segment from P, C again computes the available bandwidth for the next segment, and estimates a lower throughput due to increased round trip time (RTT) of $S(n+\alpha+1)$. Therefore, C will switch to a lower representation for the subsequent segment. This process will be repeated again and again due to Adv, hence causing the requests from C to result in many cache hits and misses rapidly.

Algorithms 2 and 3 depicts the procedure at C to stream $S(n)_{b_{i,j}}$ adaptively after an attack. Algorithm 2 illustrate the procedure, where C requests N segments in a sequential order with dynamic bitrate adaptation. The bitrate selection for each requested segment in defined in Algorithm 3.

Algorithm 2 Client (C) Segment selection process

procedure SELECT_SEGMENT_PROC($S(n)_{b_k}$,i,j,α,f)
    MPD $\leftarrow$ *Send_ requests_ to_ P* ▷ $MPD = \{S(n)_{b_{i,j}}\}$
    $S(r)_{b_k} \leftarrow$ *Select_ segment_proc()*
    $Content(S(r)_{b_k}) \leftarrow Interest(S(r)_{b_k})$
    $S(n)_{b_k} \leftarrow S(r)_{b_k}$
    $ind \leftarrow r$
    while $ind \neq N$ **do**
        $k \leftarrow$*Bitrate_adaptation_proc*$(S(n)_{b_k}, i, j, \alpha, f)$
        $Content(S(m)_{b_k}) \leftarrow Request(S(m)_{b_k})$ ▷ $m \in (r, N]$
        $S(n)_{b_k} \leftarrow S(m)_{b_k}$
        $ind \leftarrow m$
    end while
    end while
end procedure
close;

Algorithm 3 Client (C) bitrate selection process

procedure BITRATE_ADAPTATION_PROC($S(n)_{b_k}$,i,j,α,f)
    if $n == r + q \times \alpha$ **then** ▷ $q \in \{1, \ldots, (N-r)div\ \alpha\ \}$
        download_rate $\leftarrow$ *Adaptation_control_sys*()
        if $k \leqslant$*download_rate* **then**
            temp_k $\leftarrow w$ ▷ $w \in (i, j]$
        else
            temp_k $\leftarrow w$ ▷ $w \in [i, j)$
        end if
        end if
    else
        download_rate $\leftarrow$ *Adaptation_control_sys*()
        if $k \geqslant$*download_rate* **then**
            temp_k $\leftarrow w$ ▷ $w \in i$
        else
            temp_k $\leftarrow w$ ▷ $w \in (i, j]$
        end if
        end if
    end if
    end if
    Return temp_k
end procedure
close;

For each segment request, adaptation control system of DASH estimates the download rate to select the best suited bitrate. Since the download rate is affected by the RTT of segments received from CS, therefore, for the segments $S(n+\alpha+1)$, the highest bitrate is requested, i.e., b_j. However, again due to low download rate for the subsequent segment, the lower bitrate is requested.

To simplify the functioning of BOA, i.e., exploitation of interest aggregation and in-network caching, we consider a specimen of multimedia data. Let's assume that C wants to fetch a video file (V_{file}) that consists of ten segments, say $S_{(1,.,10)}$. At producer, each segment is available in multiple video bitrate ($b_{i,j}$). To launch the attack, Adv issues a sequence of interests at time t_i in an ascending order, with a consecutive gap of one. The adversary requests each segment individually in all available bitrate, particularly, $S_{1(b_{i,j})}, S_3(b_{i,j}), .., S_9(b_{i,j})$. When the edge ICN router receives interests for $S_1(b_{i,j})$, it first checks the content store (CS) for the respective segment. Since the segments are not previously cached, therefore, PIT marks the entries for the interest ($S_1(b_{i,j})$) and forwards them to FIB in order to route them to the producer (P). On the retrieval of relevant segments, all the on-path routers follow in-network caching [220], and thus stores $S_1(b_{i,j})$ in CS. In the similar way, the segments $S_3(b_{i,j}), S_5(b_{i,j})$,.., till $S_9(b_{i,j})$ are also requested by Adv and are stored in the CS of the on-path routers.

After the aforementioned process, assume that at time t_j ($t_j > t_i$), C starts streaming the same video file, i.e., V_{file}. Initially, it retrieves the MPD to get the characteristics associated with all the segments. For S_1, C requests the bitrate, say b_k by considering its own context, i.e., throughput estimation and DASH adaptation logic preferences [128]. Since all the bitrates for S_1, including b_k, are previously requested by Adv, therefore, the request for $S_1(b_k)$ is replied by the first on-path router's cache. It results in high throughput estimation for the next segment (i.e., S_2) due to reduced RTT of S_1. Due to the reduced RTT, C requests S_2 with highest bitrate, i.e., $S_2(b_j)$. However, $S_2(b_j)$ was not requested by Adv, thus it will not be available in the cache of the router. As a result, $S_2(b_j)$ is replied by the P with higher RTT. Based on the new RTT, the C again reduce the resolution and request bitrate b_k for S_3. For the full multimedia streaming, the above process of C, switching the bitrate for alternative segments between b_k and b_j repeats, thus the BOA is caused successfully.

Due to the functionality of the procedures mentioned above, C will experience undesirable bitrate oscillations, manifesting as continuous switches between high and low representations, leading to degradation in user-perceived QoE. Moreover, the playback buffer depletes in case of repeated

oscillations and forces C to take radical measures to refill it at the expense of smooth streaming. Below we present the details of our two proposed countermeasures to mitigate the BOA and provide efficient DAS: (i) a receiver driven approach in 2.4, and (ii) a robust network architecture in 2.5.

2.4 Fair-RTT-DAS: RTT Fairness for Dynamic Adaptive Streaming over ICN

In this section, we present the details of our proposed countermeasure called *Fair-RTT-DAS* that mitigates the BOA in DAS over ICN. We show that unlike the traditional Internet architecture, it is not sufficient to estimate the bitrate for next segment in ICN by just considering the RTT values of the adjacent receiving segments. This is because the producer location keeps on changing in ICN due to the content source variation caused by in-network caching and interest aggregation. It leads to the radical difference in RTTs of consecutive segments retrieved in an on-going streaming session. In general, the segment(s) retrieved from intermediate routers will have small RTT as compared to the ones received from the producer (P). If the change in the consecutive segment's location in a session is too frequent, DAS will falsely estimate a higher or lower throughput for the subsequent segments. We discussed in section 2.3 that the false throughput estimation stands as a vulnerability in DAS over ICN. We claim that in DAS over ICN, it is not sufficient just to discuss the throughput fairness which narrates the style of TCP/IP related research [42, 177, 218]. In our approach, we emphasize to maintain fairness in throughput estimation for the segments (within a single video file) with varied source locations.

To guarantee trustworthiness and evenness in bitrate adaption, we design a consumer-driven model. Our proposal preserves the fairness in the segment's RTT within a streaming session. Fair-RTT-DAS countermeasures the inference attacker with the aim to attain the following requirements.

- Significantly reduce the precision of BOA in order to effectively alleviate the adverse impacts of the attack.
- Efficient bandwidth utilization to download appropriate bitrate segments in dynamic network conditions.
- Ensure scalability in terms of the deployment of our countermeasure with lower additional overheads.

To accomplish the above objectives, Fair-RTT-DAS dynamically estimates the available bandwidth, identify the false estimations, and circumvent for the default bandwidth estimation process that is used in conventional bitrate adaptation method. Fair-RTT-DAS framework ensures fair-RTT based mechanism on top of DAS streaming control system. The significant element of Fair-RTT-DAS is its unique rate adjustment function that leads to adequate throughput estimations in presence of content source variations.

Fair-RTT-DAS entails a set of algorithms integrated to DAS client with the goal to identify the attack and moderate their marks. In addition, the intermediate routers are not required to report the accumulative statistics of inward and outward interests to DAS client. Thus, Fair-RTT-DAS has the advantage due to its ease of deployment and scalability in view of growth in network traffic. Fair-RTT-DAS consists of two major phases: (i) *detection*, and (ii) *reaction*. In the detection phase, the DAS client identifies the attack, while the reaction phase eccentrically controls the sending rate of the interest packets.

2.4.1 Basic idea of Fair-RTT-DAS

The elementary idea of Fair-RTT-DAS is to enforce the DAS client to use a collaborative approach for bitrate adaptation, which includes our definition of RTT-fairness in conjunction with conventional bandwidth estimation approach. Since conventional bitrate adaptation depends mainly on bandwidth estimations, thus it is not adequately ingenious to identify the variations in the source of the content. Hence, to keep the bitrate oscillations to the minimum and low switching amplitude [135, 154], our strategy exploits fair-RTT-based bitrate adaptations to fetch the best available bitrate representation in the presence of an adversary.

The foremost concern is to ensure that the DAS client identifies the segments of the media content with the variation in source location, during the streaming process. Secondly, bitrate adaptation process should efficiently request the available representations to avoid the false bandwidth calculations to mitigate the BOA. By careful analysis of the variations in RTT between the consecutive interests and their corresponding content of segments, Fair-RTT-DAS identifies the symptom of adversarial presence. Hence, to avoid the inferior behavior of DAS, Fair-RTT-DAS dynamically modifies the sending rate of interest messages for the segments that might cause the QoE degradation in the session.

2.4.2 Detection Phase

To maintain RTT-fairness, our model identifies the source for chunks (i.e., fragments[1]) by considering packet level communication statistics. Although ICN routers cache segments and frames independently, but DASH performs bandwidth estimation on the segment level listed in MPD, instead of the fragment level. Below we describe in detail the functionality of Fair-RTT-DAS for intra-segment streaming (i.e., at fragment level) and source variation detection.

2.4.2.1 Intra-segment Communication

Fair-RTT-DAS aims to identify the variation in content source by taking advantage of the transport and link layer statistics [184] [177] [29]. The segments in DASH stands as a sequence of bytes identified by a globally unique identifier, and these may vary with each other in size [127]. The network devices are restricted to forward packets up to the maximum transmission unit (MTU), thus segments are fragmented into smaller chunks (or frames) before transmission. The common design approach of fragmentation - as this work assumes - is that each resulting fragment of a segment represents a uniquely identifiable piece of transmission and addressed unit in its own right [29, 153, 184]. For instance, any segment larger than 1449 bytes [2] is fragmented and identified independently. When DASH client requests a segment longer than the maximum packet size of the network, the response contains the meta-data of that requested segment. The meta-data includes fragment identifiers (i.e., interests) that makes up the requested segment, segment size, and additional security and integrity information [29] [88], as it is shown in Figure 2.6. DASH client then issues pipeline of interests to request fragments simultaneously in order to completely utilize the link capacity. This is accomplished dynamically by calculating the instantaneous *interest sending rate* $x(t)$ that client handles [42], as illustrated in Equation 2.2,

$$x(t) = \frac{w(t)}{p + \frac{q(t)}{c}}. \tag{2.2}$$

[1] Please note that a video session is divided into multiple segments and a segment greater than MTU could further consist of multiple fragments.

[2] Amustndnsim currently packetizes media into segments that are less than the typical 1449 bytes.

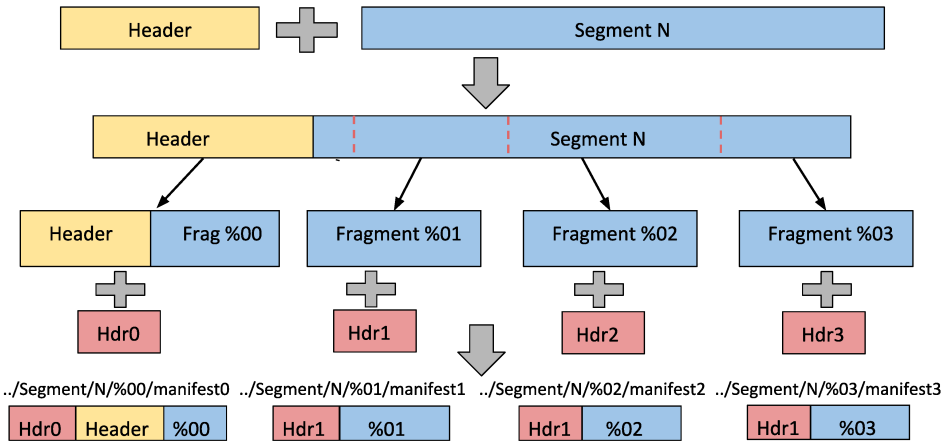

Figure 2.6: Segment fragmentation

Here, $w(t)$ defines the size of the receiver window, which is the maximum number of interest a client is allowed to send while waiting for the corresponding segments. $q(t)$ denotes the instantaneous queue occupancy at the source, c is the link capacity in packets/sec, and p is the round trip propagation delay. The equation shows that the interest rate linearly grows proportional to the inverse of round trip time RTT.

Based on the above pattern, gauging the segment request is an umbrella concept, which consists of a group of consecutive interests for fragments belonging to the same segment. Typically, there is no correlation between fragment requests related to two consecutive segments for bitrate selection directly. The DAS bitrate adaptation model selects appropriate representation for the whole next segment based on the previous segment download rate. This approach completely ignores the fragment-level effects on bitrate selection. We design a model to integrate the intra-segment communication with conventional bitrate adaptation model. For this purpose, Fair-RTT-DAS keeps several statistics on intra and inter-segment communication level. In particular, it accounts the RTT of the initial fragment (RTT') relevant to a previous segment(s). It also records the *interest sending rate* $x(t)$ used to request the fragments in the previous segment(s).

2.4.2.2 RTT Measurements

To identify the source variations throughout a session, Fair-RTT-DAS measures the time taken to send the first interest for each segment and receives the relevant content. As described earlier, this content may contain informa-

tion relevant to fragmentation and link-layer communication. The resulting rate of interest packet sending and content delivery highly depends on the notion of the distance between the client and the content source. Here we explicitly define it through *effective round trip time* (RTT'), in analogy with the RTT of connection-based transmission protocols like TCP [42].

Our mechanism implements the bootstrapping mode on DASH client, which operates by calculating the moving average of RTT' for initial fragments received from a number of previous segments in a constant time period, T. Since the proposed attack relates to the variation in the source location, Fair-RTT-DAS compares the RTT' variations with the time T, which in case of a successful attack is noticeably radical as compared to the network congestion. In addition, Fair-RTT-DAS uses average measured RTT' in each consecutive T instead of the weighted average of RTT'. It is because the weighted average is effective only in case of smooth rate adaption and removal of measuring errors. Moreover, it is more functional to eliminate the variation in long-term source variations. Therefore, the average measured RTT' for each constant time period enables the detection of attack [191]. In addition, to distinguish these reasons, i.e., the RTT' variation caused by adversary or network congestion, the statistical time unit (i.e., T) must be set to a small value.

2.4.2.3 Source Variation Detection

In our method, DASH client remains in bootstrapping mode until it executes the measurement of average RTT', namely $RTT'(T)_{avg}$ in the first T time period. After this bootstrapping mode, Fair-RTT-DAS compares the RTT' of each first fragment of the current segment with the $RTT'(T)_{avg}$ of previous segments. In case the RTT' received of the current segment is greater than $RTT'(T)_{avg}$, the detection phase leads the system to move as per DASH streaming control system. However, if the value is radically smaller w.r.t $RTT'(T)_{avg}$, this leads to identify the attack and source variation, as shown in Equation 2.3. In this way, the detection phase is able to identify source variations among consecutive segments which can trigger oscillations.

$$RTT' + jitter \lll RTT'(T)_{avg} \tag{2.3}$$

To better illustrate the bootstrap phase, we designed a model in Algorithm 4, which stores the RTT' of the manifest file of each segment called $manifest(S(r)_{b_k})$ that were received in during last T time period. In particular, the first fragment to request is the file containing meta-data (i.e., mani-

Algorithm 4 Fair-RTT-DAS algorithm (Bootstrap phase)

procedure SELECT_SEGMENT_PROC($S(n)_{b_k}$,i,j,α,f,T)
  $MPD \leftarrow Send\_\ requests\_\ to\_\ P$ ▷ $MPD = \{S(n)_{b_{i,j}}\}$
  $S(r)_{b_k} \leftarrow Select\_\ segment\_proc()$
  $Content(S(r)_{b_k}) \leftarrow Interest(S(r)_{b_k})$
  $Time\_queue := empty$
  $RTT\_queue := empty$
  $Flag : false$
  while $(Time\_queue.sum() \leq T)\&(r \neq N)$ **do**
    $k \leftarrow Bitrate\_adaptation\_proc(S(r)_{b_k}, i, j, \alpha, f)$
    $t\_end(S(r)_{b_k}, t\_start(S(r)_{b_k}, RTT'\_manifest(S(r)_{b_k}), S(m)_{b_k} \leftarrow Request\_fragment(Flag, Time\_queue, RTT\_queue)$
    $Time\_queue\_add(t\_end(S(r)_{b_k}\text{-}t\_start(S(r)_{b_k})$
    $RTT\_queue\_add(RTT'\_manifest(S(r)_{b_k})$
    $S(r)_{b_k} \leftarrow S(m)_{b_k}$ ▷ $m \in (r, N]$
  end while
  end while
  $Flag : true$
  while $r \neq N$ **do**
    $k \leftarrow Bitrate\_adaptation\_proc(S(r)_{b_k}, i, j, \alpha, f)$
    $t\_end(S(r)_{b_k}, t\_start(S(r)_{b_k}, RTT'\_manifest(S(r)_{b_k}), S(m)_{b_k} \leftarrow Request\_fragment(Flag, Time\_queue, RTT\_queue)$
    $Time\_queue\_add(t\_end(S(r)_{b_k}\text{-}t\_start(S(r)_{b_k})$
    $RTT\_queue\_add(RTT'\_manifest(S(r)_{b_k})$
    if $Time\_queue.sum() \geq T$ **then**
      $Time\_queue.PoP()$
      $RTT\_queue.PoP()$
    else
      $Flag : false$
    end if
    end if
  end while
  end while
end procedure
close;

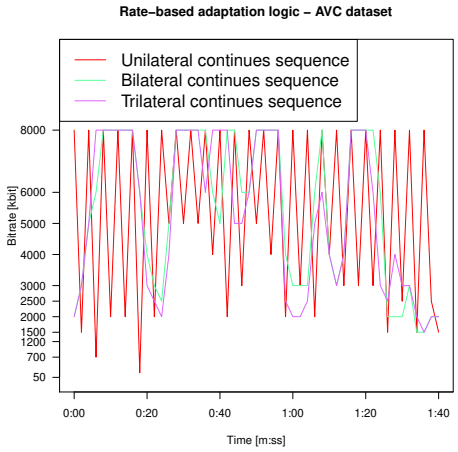

Figure 2.7: Comparison of attack sequences (RB-AVC)

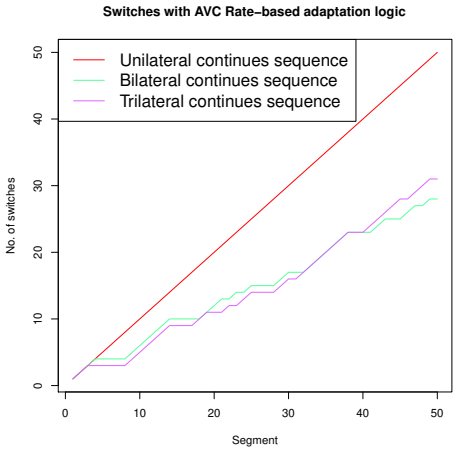

Figure 2.8: Switches comparison of attack sequences (RB-AVC)

fest file). While streaming the session, if the time value ($Time\_queue$) is less than T, Fair-RTT-DAS selects the b_k as per default bandwidth estimation. Therefore, DAS streaming control system selects the appropriate bitrate as shown in Algorithm 3. Furthermore, the interest packet sending rate $x(t)$ is maintained by Equation 2.2. Throughout the streaming, the T (i.e., the time in which average RTT' (RTT_queue) is measured) slides along the session and it always stores the $RTT'(T)_{avg}\_manifest(S(r)_{b_k})$ for previous T. In particular, Fair-RTT-DAS uses moving average $RTT'(T)_{avg}$ in each consecutive T, which is always measured dynamically at each instance for previous T period. Therefore, the moving average measured ($RTT'(T)_{avg}$) for each constant time period enables the detection of attack or source variation in an ongoing streaming session. The proposed novel fragment request procedure is described in Algorithm 5, which operates during bootstrap-phase and after it.

2.4.2.4 Parameter setting

Our investigation shows that maximum oscillations during the attack only happens when an attacker generates interests with consecutive gaps. In case that attacker issues two contiguous interests with a consecutive gap, the efficiency of the attack reduces by 50%. In fact, adversary helps the victim to improve its QoE, if it requests more than two continuous segments. Since in this case, the client will be receiving most of the continuous segments from

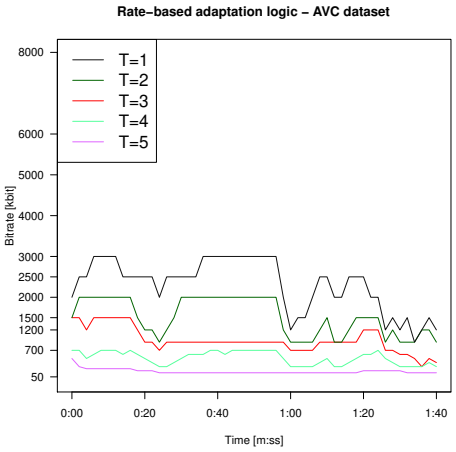

Figure 2.9: Bandwidth and T relation (RB-AVC)

the edge or intermediate routers, resulting in reduced bitrate oscillations and high bandwidth utilization. Therefore in our experimental analysis, adversary requests a unilateral sequence of segments to achieve the maximum efficiency in the attack. Figure 2.7 and 2.8 show the relation of switching frequency and number of continuous segments requested by the (an) attacker. From the results, it can be seen that switching frequency declines when *Adv* requests more number of consecutive segments.

The value of the time period (T) depends highly on user-end application and service requirements for a streaming session. The higher value of T may improve the smooth bitrate selection, but it can be a factor against efficient bandwidth utilization. The value of T also relates to the number of segments issued continuously by an attacker. It is because the detection phase uses the value of T to detect the RTT' variations. In addition, the number of segments falling in the T period helps reaction phase to identify the positive cache hit which may help to increase the user's QoE. For instance, if many continuous segments are available on the router's cache due to content popularity, the reaction phase will efficiently utilize the bandwidth and switch to higher bitrates (later detailed in Section 2.4.3).

The simulation results in Figures 2.7 and 2.8 reports that maximum impact of attack is seen at times when adversary generates unilateral continuous segment with consecutive gap. Therefore, if the attacker is issuing the request to generate maximum attack impact, than the minimum number of segments to fully detect the attack should be more than one. In our case, the most appropriate value of T is at least equal to the playback duration

of two segments. In addition, the relationship of T with the bandwidth utilization is shown in Figure 2.9, and the results report that for increased value of T, the client exhibits smooth rate adaptation and fewer switches. Although, it remarkably reduces the bandwidth utilization and streams with lower bitrate values.

2.4.3 Reaction Phase

Once an ongoing attack is suspected, it triggers the Fair-RTT-DAS to react. To have a smooth rate adaptation, we name this process as *adaptive phase*, and it aims to adopt best possible bitrate in the presence of an adversary. For each segment after the detection, the countermeasure adaptively increases the interval time between the fragment requests belonging to the segments. In general, client sends interest fragments according to the sending interval $d(t)$ (please refer to Equation 2.4). In Equation 2.4, k (refer to Equation 2.5) is the correction of data packet size; s (in bytes) is the maximum chunk size (i.e, MTU), and u is the pre-defined constant value [191]. The size of the data packet (s) is highly dependent on the individual application and network. Also, the bandwidth consumption of their data packets are different when each application send s interests with the same $x(t)$. Therefore, we use k to remove the difference in data packet size among competing data packet flows. For instance, $d(t)$ will increase with the growth of s as compared with other flows. In our model, we take s uniform in the whole scenario. Hence, to mitigate the attack, the adaptive phase decreases $x(t)$ for each identified segment that increases the interval between the interests sent for fetching the fragments $d(t)$.

$$d(t) = \frac{1}{k * x(t)}; \tag{2.4}$$

$$k = \frac{u}{s}. \tag{2.5}$$

The functionality of the bootstrap phase is to process according to the DAS steaming control system and to select the $x(t)$ according to default streaming system. In case of a successful attack while bootstrapping is ongoing, it may seem to have false bandwidth estimation. However, it is required since to obtain $RTT'(T)_{avg}$. After the bootstrap phase, if the algorithm detects the attack, the DAS client adopts to the adaptive phase. The key contribution of the adaptive phase is to reduce interest sending rate ($x(t)$) after attack identification for the future segment. Hence, the bitrate estimation done by DASH streaming control system is directly affected by controlling $x(t)$ to have better QoE. It functions and controls the fragment

sending rate in order to maintain evenness in downloading rate of identified segments comparing to previous ones.

In case of an attack, when a fragment is replied back from the *router* instead of the *producer*, RTT' decreases drastically. The adaptive phase will decrease the interest sending rate by increasing the $d(t)$ for the remaining fragments within that specific segment. It leads to the reduced download rate, although, the segment is replied by the router's cache. Here the parameters to control the $x(t)$ are made directly proportional to QoE perceived by the user. Therefore, the adaptive phase maintains the inter-packet interval, which will be the same as it is experienced by the previous segment. To accomplish this, the DAS client maintains $d(t)$ for current segment using the previous segments $x(t)$ values.

Fair-RTT-DAS performs equally better in case of content source variation. For instance, if most of the continuous segments are stored on routers, it will help to increase the QoE for default dynamic adaptive streaming. It is due the advantage of two fundamental aspects which Fair-RTT-DAS takes into account. First, moving average RTT calculation of initial fragment for each segment in T period, $(RTT'(T)_{avg})$. Secondly, smooth bitrate adaptation which also considers the positive cache hit for efficient bandwidth utilization. Therefore, considering the value of T as a playback duration of two segments and its sliding nature, the $RTT'(T)_{avg}$ always reduces when subsequent segments are being retrieved from the router's cache instead of the producer. Later, this new reduced value of $RTT'(T)_{avg}$ will be used to detect the source variation, which will help to increase the bitrate. For instance, if more subsequent segments results in lower RTT', the adaptive phase selects higher interest sending rate, by maintaining the packet interval of previous segment $d(t)$, which is retrieved from the router's cache. It is worth mentioning here that we have selected the value of T equal to the playback duration of two segments. Hence, if the subsequent segments are coming from routers cache, e.g., with reduced RTT', the adaptive phase will select previous $d(t)$ value, which is more reduced. In this way, the interest sending rate $(x(t))$ is increased for the current segment, and it results in increased bitrate adaptation in case more subsequent segments are available in caches, and it helps to increase the bandwidth availability for DASH client.

The detailed description of the adaptive and detection phase is given in Algorithm 5. To detect, it compares the RTT' of manifest of each segment $(RTT'\_manifest(S(r)_{b_k}))$ with the average $RTT'(T)_{avg}\_manifest(S(r)_{b_k})$ which is estimated over time T. In case of an attack, the DASH client reduces the $x(t)$ to the value of previous segment. This leads to increase in the inter-fragment interval $(d(t))$ within a segment. Conversely, in order to

Algorithm 5 Client (C) Fragment request algorithm (Detection and Reaction phase)

procedure REQUEST_FRAGMENT_PROC($S(r)_{b_k}$,Flag, Time_queue,RTT_queue)
  $Content(S(r)_{b_k}) \leftarrow Request\_manifest(S(r)_{b_k})$
  if Flag = true **then**
    if $RTT'\_manifest(S(r)_{b_k} \lll RTT\_queue.sum \div RTT\_queue.length()$ **then**
      $x(t)\_(S(r)_{b_k}) \leftarrow x(t)\_(S(r)_{b_k-1})$
      $Content\_fragment_i\_(S(r)_{b_k}) \leftarrow Content\_fragment_i\_(S(m)_{b_k})$ ▷ $i \in (1,n], m \in (r,N]$
    else
      $DASH\_control\_sys() \leftarrow x(t)\_(S(r)_{b_k})$
      $Content\_fragment_i\_(S(r)_{b_k}) \leftarrow Content\_fragment_i\_(S(m)_{b_k})$ ▷ $i \in (1,n], m \in (r,N]$
    end if
    end if
  else
    $DASH\_control\_sys() \leftarrow x(t)\_(S(r)_{b_k})$
    $Content\_fragment_i\_(S(r)_{b_k}) \leftarrow Content\_fragment_i\_(S(m)_{b_k})$ ▷ $i \in (1,n], m \in (r,N]$
  end if
  end if
end procedure
close;

make client compatible with network congestion and to increase bandwidth, the fragments are sent as per default DASH streaming control system, i.e., instantaneous *interest sending rate* $x(t)$ that client handles [42] (refer to Equation 2.2).

2.4.4 Evaluation and Result Analysis

In this section, we investigate the performance of adaptive multimedia streaming over ICN in presence of adversary. We implement and evaluate the effectiveness of our proposed countermeasure namely Fair-RTT-DAS for multimedia streaming over DASH in ICN. To this end, we perform extensive simulations using AMuSt-ndnSIM, which is an Adaptive Multimedia Streaming Framework for ndnSIM [116]. AMuSt-ndnSIM framework provides support to create a bridge between multimedia traffic and NDN [222], categorically based on ndnSIM [20] [139] and libdash [148]. Note that NDN is a specific instantiation of ICN which is well-suited for this evaluation. AMuSt framework offers a set of applications for producing and consuming adaptive video traffic but exchanging HTTP with NDN. The functionality of DASH is provided by the libdash library, which is an open source library with an interface to DASH standard and an official reference software for DASH standard [148].

2.4.4.1 Test Setup

To set up the tests, we implement the network depicted in Figure 2.4 with a single origin server and a number of multimedia clients (including honest and malicious hosts) connected with multiple NDN routers. To configure the producer (P) with real-time video traffic, we use an AVC-encoded multimedia video [126] and the *BigBuckBunny* movie from the DASH/SVC Dataset [115]. Other network parameters and their values used in our test setup are given in Table 2.2. The forwarding strategy used at NDN routers is minimum hop count *(BestRoute)*, and we chose *Least Recently Used* (LRU) as a caching policy for router caches.

2.4.4.2 Evaluation Metrics

The QoE in video streaming relies on the intermingling of high video quality (e.g., high bitrate) and high streaming performance (e.g., continuous playback without re-buffering). The authors in [137] and [135] explain the impairment factors that affect the user experience for dash video, and it illustrates that frequent switching in video representations in a session

Table 2.2: Parameters for simulations

Parameters	Value
No. of video segments (N)	250
Video period(s)	100
No. of edge routers	1
Available bitrates (AVC)	20
Layers of quality (SVC)	4
Duration per segment(s)	2
Bandwidth between the nodes (Mbps)	10
Delay between C to edge router (µs)	200
Point to point delay (µs)	10
Max buffer size (s)	30
Consecutive gap α	2
fragment size s (byte)	1449
Constant value u	1449
Value of T (s)	3
Drop Tail Queue (max. packets)	20
Cache policy	LRU
Start up delay (s)	0.1
Max. buffered time (s)	30

diminishes streaming quality. Thus, the spatial quality of video can be determined by the level of variations occurred during a streaming session. To evaluate the attack and Fair-RTT-DAS, we use the following metrics.

- Number of switches: It indicates the frequency of video quality switches [154];
- Average switch magnitude: It indicates the average amplitude of the video quality switches [135, 154].

2.4.4.3 Attack Impact

Figures 2.10 and 2.11 show that *Adv* is able to degrades the QoE of client while being connected to any on-path router, however, the maximum degradation occurs when it is connected to the same first-hop router. In simulation results, we report the case when *Adv* is connected to the same first-hop router to which the client is connected. Figures 2.12 and 2.13 report the bitrate requested by the DAS client for both the cases (with and without the attack) using RB and R&B adaptation logic. Our results in Figures 2.15

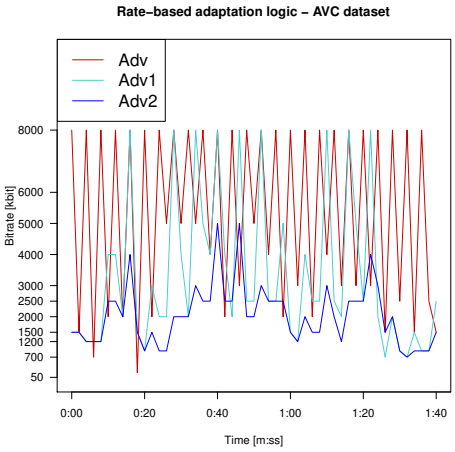

Figure 2.10: Dynamic adaptive streaming to different adversarial locations using RB (AVC)

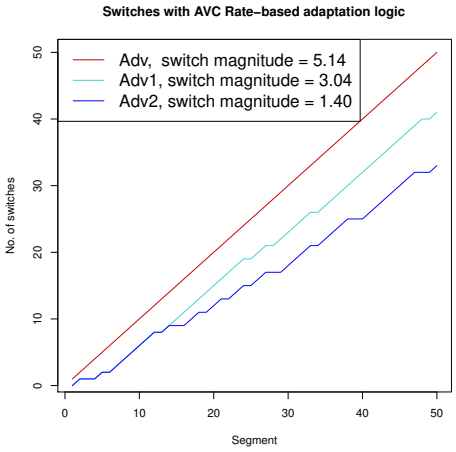

Figure 2.11: # of switches to different adversarial locations using RB (AVC)

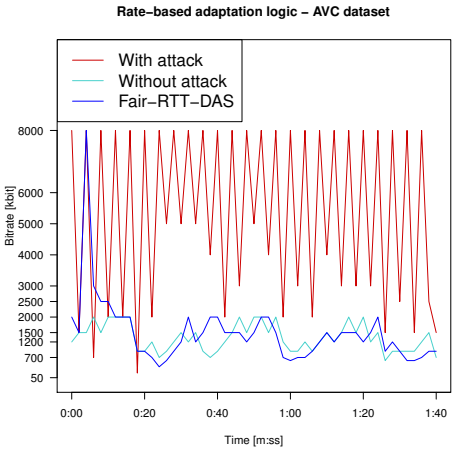

Figure 2.12: Dynamic adaptive streaming using RB (AVC)

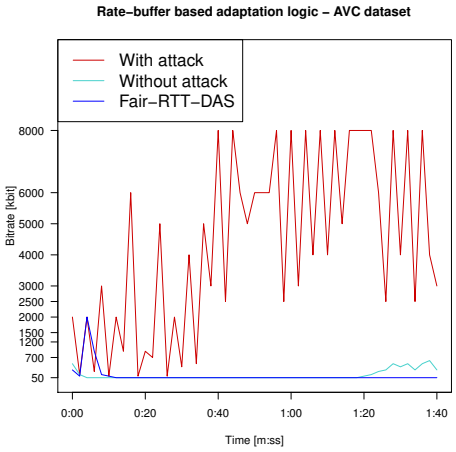

Figure 2.13: Dynamic adaptive streaming using R&B (AVC)

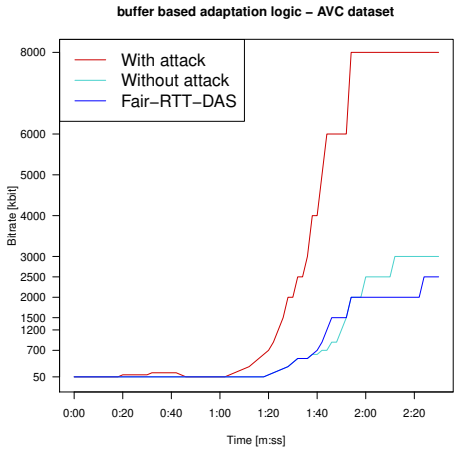

Figure 2.14: Dynamic adaptive streaming using BB (AVC)

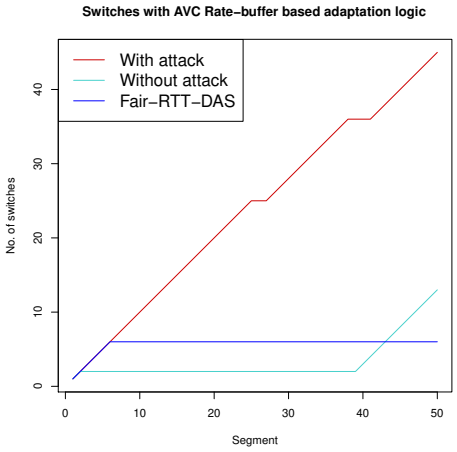

Figure 2.15: # of switches RB (AVC)

and 2.16 show that frequency of bitrate switching in RB and R&B adaptation logic increases remarkably in the presence of an adversary. Moreover, the attack massively increases the average switch magnitude of bitrate fluctuations for these adaptation logic while streaming AVC content (please refer to Figure 2.22).

The results in Figure 2.14 shows the bitrate requests in BB adaptation logic (AVC). Figure 2.17 depicts that the DASH client experiences an increase in the number of switches, and it can be seen at first glance that a victim is experiencing higher video bitrate. However, considering the user's QoE evaluation metrics (detailed in Section 2.4.4.2), this not satisfactory. From Figure 2.17, we can see that the number of switches in Buffer-based (AVC) are much higher in presence of an adversary for default DASH over ICN. But, with Fair-RTT-DAS, it is merely equal to scenario without attack. Similarly, we can see the average switch magnitude difference in Figure 2.22, which shows that without our proposed approach the user is experiencing approximately 40% more average switch magnitude (i.e., 0.3 with attack, and 0.18 with Fair-RTT-DAS). From these two metrics, we can observe a remarkable QoE degradation for users in the presence of attack, and Fair-RTT-DAS is able to mitigate the attack as well as it performs efficiently as compared to normal scenario (i.e., without attack).

Categorically, for all adaptation logic in AVC, Figure 2.22 shows that the client experiences an increase in average switch magnitude of bitrate fluctuations in the presence of the attacker. This increased switch magnitude

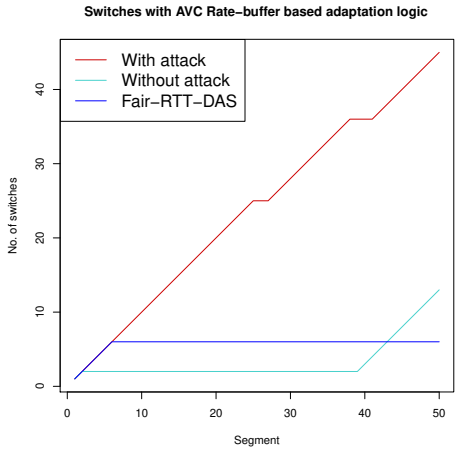

Figure 2.16: # of switches R&B (AVC)

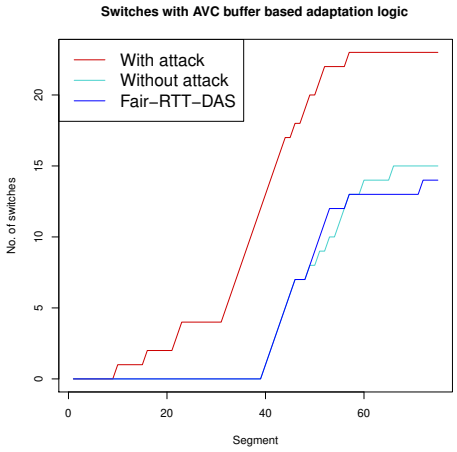

Figure 2.17: # of switches BB (AVC)

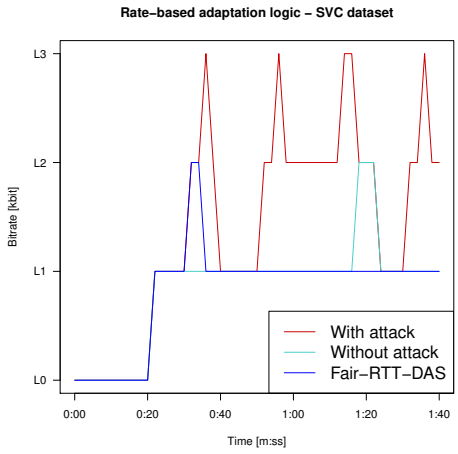

Figure 2.18: Dynamic adaptive streaming using RB (SVC)

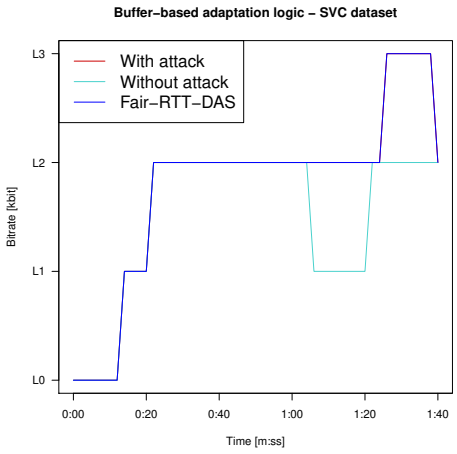

Figure 2.19: Dynamic adaptive streaming using BB (SVC)

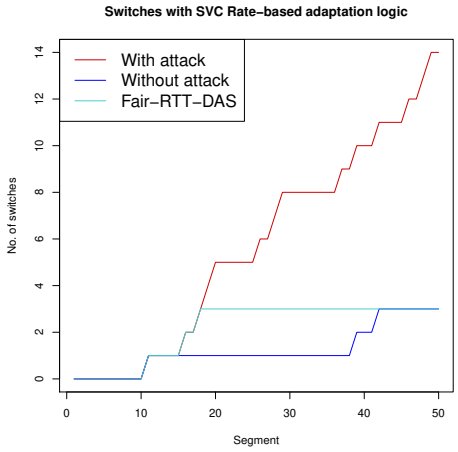

Figure 2.20: # of switches RB (SVC)

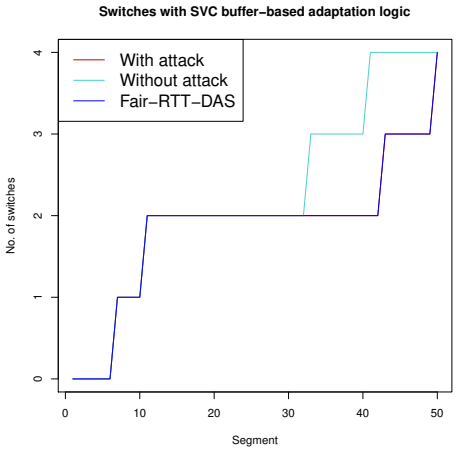

Figure 2.21: # of switches BB (SVC)

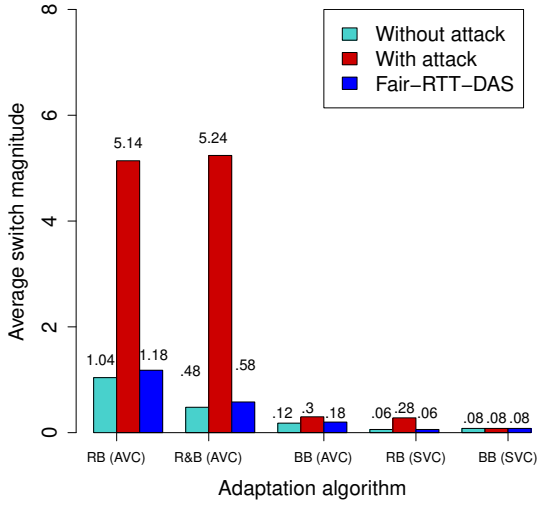

Figure 2.22: Average switch magnitude

is experienced by client because the adversary forces the client to switch multiple times and between extremely low and high resolutions.

For the RB adaptation logic in the SVC dataset, the attack results in higher frequency of switches in the download bitrate (refer to figures 2.18 and 2.20). However, the magnitude of the switches is relatively small when compared to what we found for our AVC dataset as it is shown in Figure 2.22. It is because the number of available layers of representations (i.e., three EL and one BL) is low as compared to the twenty representations available in AVC. Regardless, the adversary is able to cause a higher number of bitrate switches in SVC dataset with respect to normal conditions, leading to a reasonable QoE degradation. From the simulations, we also observe that buffer-based adaptation logic in SVC dataset is unaffected by the attack. Figures 2.19 and 2.21 show that there is no increase in bitrate fluctuations and average switch magnitudes. We identify that buffer capacity affects positively and resits to short-term bandwidth fluctuations. However, use of the buffer based adaptation logic still remains an open question for researchers due to buffer size management in relation to the playback time, since in large networks multimedia delivery imposes dramatic burden on in-network caching.

In our simulation setup, we consider the case of a single DASH client in the presence of one or more *Adv*. It is because we focus on the attack, which is subjective to the vulnerability identified in DASH bitrate adaptation logic. The case in which more than one victim may attach to the same edge router to which *Adv* is connected also faces the similar QoE degradation. However, it entails two cases: (i) the bitrates requested by all the victims solely depends on their bitrate adaptation strategies and hardware constraints. Besides, all users might face different bandwidth fluctuations considering different access networks, e.g., WiFi, LAN, 3G, 4G etc. Therefore, it is not guaranteed that all the users will request similar video bitrates for the same multimedia content. In this case, all the victims will face the related impact of attack since all clients will counter the specific interest sequence generated by *Adv*, and (ii) if there are numerous victims attached to the same first-hop router and requests the same multimedia content, it may result in content popularity for that content since all tend to request the segments in a sequential manner with different bitrates. Due to an increased number of clients requesting the same segments also increases the chances of same bitrates requested by the clients. Therefore, considering the efficacy of in-network caching, most of the continuous segments with multiple bitrates gets stored on the routers. It helps the bandwidth constraints and improves

user's QoE since most of the consecutive segments with various bitrates can be fetched from routers instead of the origin server.

2.4.4.4 Fair-RTT-DAS Effectiveness

The simulation results for our proposed countermeasure reports that DASH client is able to sustain the perceived QoE in the presence of an adversary. This is because Fair-RTT-DAS maintains RTT smoothing within the packets of the same video session. Our approach identifies the source variations which are hard to identify for a DASH client. After identifying the attack, Fair-RTT-DAS makes the DASH client to select the most appropriate bitrate with respect to the best possible QoE perceived. Fair-RTT-DAS also satisfies the fundamental characteristics of adaptive streaming, since DASH client adaptively adjusts the bitrate representations while effectively utilizing the bandwidth in fluctuating network conditions.

Figures 2.12 and 2.13 report the performance of Fair-RTT-DAS with and without adversarial model using the RB and R&B adaptation logic. It highlights the phenomena in which the clients follow the victim's pattern initially, which indeed represents the bootstrap phase. However, later it rapidly implements smooth bitrate adaptation. The results in Figures 2.15 and 2.16 illustrates that frequency of bitrate switching in above mentioned adaptation logic declines remarkably in the presence of an adversary. In addition, we notice a slight improvement in bitrate fluctuations comparing to traditional DASH due to bootstrapping phase. However, for some extent Fair-RTT-DAS compromises on bandwidth utilization in order to deliver smooth bitrate adaptation. The Fair-RTT-DAS correspondingly upholds the average switch magnitude of the bitrate fluctuations to default values for the adaptation logic (please refer to Figure 2.22). Results in Figure 2.14 show the bitrate requests for Fair-RTT-DAS in BB adaptation logic (AVC) and Figure 2.17 confirms the reduction in the switching frequency. Figure 2.22 confirms that the DASH client experiences merely equal value to default for average switch magnitude of the bitrate fluctuations in the presence of an attacker for all three examples of adaptation logic.

Fair-RTT-DAS results in reduced bitrate oscillations (refer to Figures 2.18 and 2.20) for the RB adaptation in SVC dataset. Moreover, it also maintains the equivalent magnitude for a number of switches when compared to the conventional DASH streaming system as it is shown in Figure 2.22. From Figures 2.19 and 2.21, we can identify that BB adaptation logic in SVC dataset takes advantage of the attack. The client may seems to have better resolution due to pre-fetching and it takes advantage of buffer

size and SVC. However, the Fair-RTT-DAS is also unaffected in this scenario and participate to enhance the victim's perceived QoE.

2.5 An Architecture for Efficient and Robust Dynamic Adaptive Streaming over ICN

In this section, we broaden the BOA attack scenarios, and subsequently propose a network based mitigation approach that is both effective and efficient. In particular, ICN's autonomous on-path cache management initiates enormous cache redundancy, results in sub-optimal selection of cached contents, and inherits network-wide cache-ignorant routing. This makes DAS more challenging in ICN by exposing it to new security risks. Therefore, we propose to mitigate BOA based on timely and global knowledge of content access information. This enables DAS to realize network-wide caching goals and cache-aware routing.

Our contribution in this work is twofold. First, we propose an effective countermeasure to mitigate BOA, called CoMon-DAS. It implements **Co**ordination with lightweight **Mon**itoring for DAS to enable network-wide coordinated caching and cache-aware routing. By this, it aims to reduce bitrate oscillations and cache content redundancy in presence of both BOA and inherent content source variations, thus to enhance perceived QoE. Second, we evaluate BOA and CoMon-DAS, through an extensive simulation study. Our results show the adverse impact of BOA, as well as the high effectiveness and feasiblity of CoMon-DAS.

2.5.1 CoMon-DAS: Coordinated Caching and Cache-Aware Routing for DAS

We aim to mitigate BOA in an effective, yet inexpensive, way. Our experience in defending against distributed attacks in ICN [173, 174, 176] learned us that effectiveness can be achieved if the attacks are mitigated based on current, network-wide view of attack-related information.

For DAS over ICN, unlike native ICN's autonomous on-path cache management scheme, caching decisions should be made based on network-wide knowledge of content requests, and routing should be aware of cache configurations (i.e., which contents are cached and, in which routers). This is because DAS client estimates the bitrate(s) of the subsequent segment(s) just by considering the measurements of the earlier received segment(s). Also, the producer location keeps on shifting due to content source variations triggered by on-path caching. Categorically, the segment(s) retrieved

from various caches results in different measurements as compared to the ones received from the origin producer. If the change in the positions of consecutive segments within a session is too numerous, DAS erroneously adopts a higher or lower bitrates for the subsequent segments. This leads to a vulnerability in DAS over NDN, as discussed in Section 2.3.

To be robust against BOA and varied content source locations, the network should effectively utilize the global cache capacity, i.e., reduce bitrate oscillations and avoid cached content redundancy, thus efficiently deliver the best possible perceived QoE. The obligation indicates that each network node should have a timely and network-wide level view of cached content information. However, such a solution requires to exchange and process massive amounts of information very frequently.

We propose to address the aforementioned problem by adapting CoMon, our framework for **Co**ordination with lightweight **Mon**itoring. Our choice is influenced by CoMon's demonstrated ability to address two similar problems in NDN: (i) network-wide cache coordination [175] and (ii) mitigation of distributed DoS attacks [173, 174, 176].

We call our solution CoMon-DAS. We give an overview of the system architecture and monitoring techniques in Subsection 2.5.2.[3] Next, we describe the defense mechanism in Subsection 2.5.3.

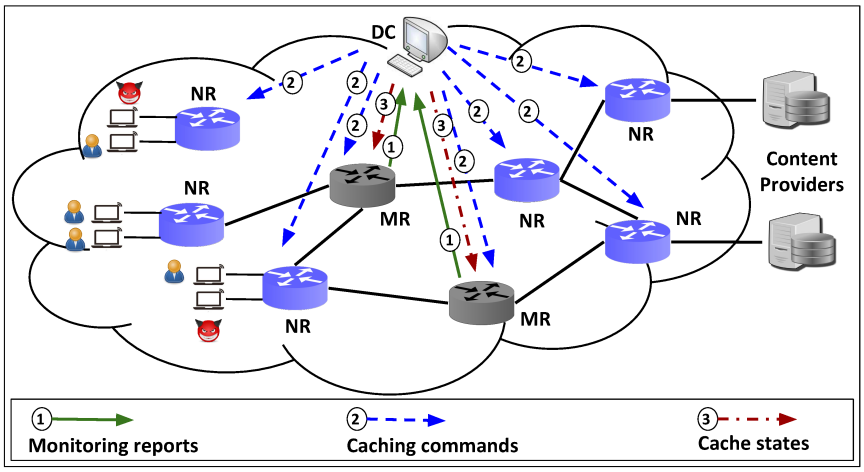

Figure 2.23: System architecture (adapted from [174]): "DC" stands for Domain Controller, "NR" for ICN Router, and "MR" for Monitoring Router.

[3] For more details about the monitoring techniques, the reader is referred to our previous work [173].

2.5.2 System Architecture and Monitoring Techniques

System architecture: CoMon-DAS is designed to work within a domain network (i.e., autonomous system). As shown in Figure 2.23, the network includes a Domain Controller (DC) and a set V of routers divided into two groups: (i) ICN Routers (NRs) and (ii) Monitoring Routers (MRs). In the following, we introduce these components and describe how do they work with each others:

1. **Domain Controller (DC)**: This is a (logically) centralized controller. It periodically receives a summary of MRs observations. The DC aggregates and processes these information. It then commands the routers to perform certain actions accordingly.

2. **ICN Routers (NRs)**: They work similar to standard ICN routers, as described in [222]. However, the routing protocol and the cache replacement algorithm are modified in the NRs.

3. **Monitoring Routers (MRs)**: Each MR, in addition to the routing and caching tasks, persistently monitors the interest packets passing through it. At the end of each observation period, the MR sends a report to the DC summarizing the names of the requested segments along with information about their quality and request statistics. The MRs also receive instructions from the DC, and adapts its routing and caching decisions accordingly.

Monitoring techniques: CoMon-DAS employs a lightweight algorithm called *PRCS* (Placement based on covered Routes and Closeness to Sources) [173] to select the MRs. In principle, PRCS selects a subset $M \subset V$ of routers[4] that jointly maximize routes coverage. At the same time, it gives preference to the routers located close to clients (thus to potential attack sources), so that attacks can be defended at an early stage.

In order to achieve full coverage, which cannot be guaranteed by PRCS alone, CoMon-DAS implements two monitoring techniques: (i) Forward-Till-Be-Monitored (FTBM) and (ii) Monitor-Aware Routing (MAR). FTBM deals with the satisfied interest packets that are not monitored earlier. Its functionality requires to add two flags to the standard interest packet: (i) *satisfied* flag and (ii) *monitored* flag.[5] When a router satisfies an interest

[4] M is predetermined; $|M| \ll |V|$

[5] The two flags neither significantly change the packet structure (only one bit each) nor breach the standard protocol.

packet [6], it sets the *satisfied* field, and then forwards the packet to the closest MR. The designated MR, in turn, records the packet information, and drops it afterwards.

MAR enforces each interest packet, thus the corresponding data packet, to pass through an MR. This requires to modify the original routing protocol. More specifically, each interest packet is first forwarded to an MR (e.g., the closest one). The designated MR then forwards the packet to its original destination.

2.5.3 Defense Mechanism

Our defense mechanism is composed of three techniques: (i) selection of cached contents, (ii) traffic shaping, and (iii) dynamic prefetching. In the following, we describe these techniques and explain how they together enable to mitigate BOA effectively.

Selection of cached contents: At the end of each observation period, the DC uses the reports received from the MRs to identify $|V| \times c$ segments to be cached in the network during the next observation window, where c denotes the router's cache capacity. Instead of assigning the segments to the routers randomly, which results in low routing performance [175], the DC considers the topological properties of the routers. In particular, it implements the allocation algorithm that we proposed in [175], which is based on the betweenness centrality (BC).

The DC makes the caching decisions corresponding to network-wide caching goals co-related with DAS requirements. In particular, a segment is cached only if it is favorable to perceived QoE. To this end, the DC exploits the aggregated report comprising the list of currently requested segments into the network, denoted as L. Using the naming information[7] from URI structure [125], the DC determines the sequence and characteristics of the segments being requested in a given period of time, δ_t. Then, for each request in L, the DC checks whether a request for a subsequent segment with higher bitrate(s) also exists in the list. If so, the DC instructs to cache the segments of both requests. Otherwise, the request is left untreated for caching.

The procedure is outlined in the first part of Algorithm 6 (lines 5 – 11). In particular, the DC checks for each request $S(r)_{b_i}$, whether the subsequent request with higher bitrate (i.e., $S(r+1)_{b_k}$) exists in L, where $i < k \leq j$. If

[6] The packet matches either a PIT entry or a cached segment.

[7] The naming information are inferred from the MPD file as part of setup phase of DASH stream.

so, both $S(r)_{b_i}$ and $S(r+1)_{b_k}$ are chased. Otherwise, the segment $S(r)_{b_i}$ is not cached. The cache assignments are then sent to the routers according to their BC values , denoted by NR_{BC} (line 12). The above described procedure restricts the non-sequential series of requests triggered by BOA to be cached. It also avoids the bitrate oscillations caused by varying content source locations.

In summary, the DC determines both: (i) the segments to be cached in the network, and (ii) their positions in the network. Those decisions explicitly enforce that contents are cached only if they are favorable to perceived QoE and reduce the redundancy degree for each selected content. The above-described procedure implies that the DC hold complete control over the entire caching process, which makes CoMon compatible with network-wide goals for DAS.

Algorithm 6 Defense mechanism against BOA

```
1: procedure FUNCTIONALITY_OF_DC (L, δ_t, S(n)b, i, j)
2:     L ← δ_t
3:     {S(n)_{b_{i,j}}} ← MPD
4:     Check requested content
5:     for Each request S(r)_{b_i} do
6:         if S(r+1)_{b_k} == L then                          ▷ i < k ⩽ j
7:             Cache the content S(r)_{b_i} and S(r+1)_{b_k}
8:         else
9:             Do not cache S(r)_{b_i}
10:        end if
11:        end if
12:    end for
13:    end for
14:    NR_BC ← Assign cache configurations
15:    Prefetching after δ_t
16:    if S(r)_{b_i} == L then
17:        Content(S(r+1)_{b_k}) ← Interest(S(r+1)_{b_k})     ▷ i < k ⩽ j
18:        NR_BC ← Instructions
19:    end if
20:    end if
21: end procedure
22: close;
```

Traffic shaping: The DC informs the MRs about caching decisions, i.e., the *segment-to-router* assignments. The MRs use these information every time they receive an interest packet not monitored before (i.e., *monitored*

= 0) to check whether the requested segment is cached inside the network or not.

Each MR, when receiving an interest packet not monitored before, checks whether both (i) the requested segment and (ii) its successor with a higher quality are cached. If so, the MR reroutes the interest packet towards the cached copy. Otherwise, the original route is preserved. Next, the MR sets the *monitored* flag to avoid repeating the aforementioned checking step by other MRs. This way, CoMon-DAS enables *cache-aware routing*.

With the above described caching and routing strategies, a request from the client results in a *cache-hit* only if the CS is capable to compete the DAS bandwidth estimation requirements for the subsequent segment. This is driven by the fact that DAS clients utilize the bandwidth estimation of the retrieved segment while processing the bitrate selection process of the subsequent segment. This way, CoMon-DAS shapes the requests to avoid bitrate oscillations triggered by BOA and varying content source locations.

Dynamic prefetching: To improve the QoE further, CoMon-DAS additionally takes proactive measures while effectively utilizing the global cache capacity. Specifically, based on the available timely content request information, the DC predicts the contents to be requested by the DAS clients in the near future. More precisely, after each observation period, the DC monitors the reports (i.e., content request information) received from MRs, and makes prefetching decision based on the segments that are already requested by the clients.

As outlined in Algorithm 6 (lines 14 – 17), the DC decides to prefetch a segment $S(r+1)_{b_k}$ if a request of $S(r)_{b_i}$ already exists in the requested content list, where $i < k \leq j$. In order to retrieve URI structure and information regarding available segments and bitrates, i.e., $S(n)_{b(i,j)}$, DC may obtain MPD information by explicitly requesting the MPD from producer. Moreover, the prefetching assignment is performed at the end of the observation period (window) and for only each subsequent segment,which is defined as a round. Subsequently, the DC assigns the prefetching tasks to the routers that are located close to the clients[8].

2.5.4 Evaluation and Result Analysis

In this section, we evaluate the BOA attack as well as CoMon-DAS. The evaluation is established on simulations. We describe our experimental setup and evaluation metrics in Subsection 2.5.4.1. After that, we discuss the

[8] The routers are selected by the algorithm that we proposed in [175].

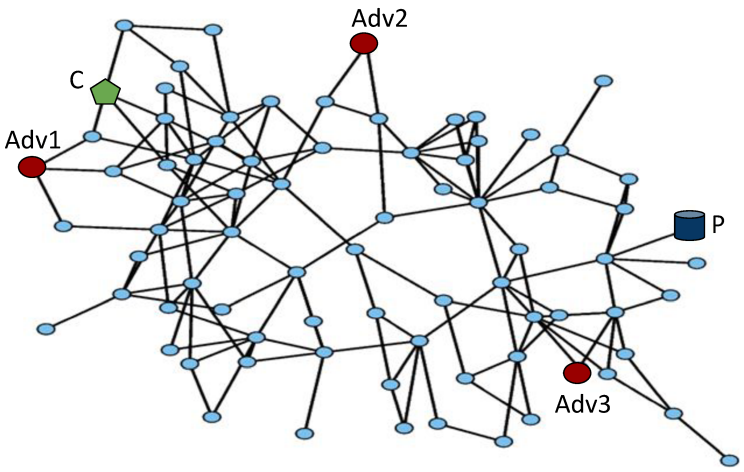

Figure 2.24: AS 3967 topology: 79 nodes and 147 edges

impact of the attack and the effectiveness of CoMon-DAS in Subsection 2.5.4.2 and Subsection 2.5.4.3, respectively.

2.5.4.1 Setup and Evaluation Metrics

We implemented BOA and CoMon-DAS over AMuSt-ndnSIM [116], an adaptive multimedia streaming framework over ndnSIM. AMuSt-ndnSIM delivers a set of applications grounded on the official DASH standard [139].

We simulated with a real ISP toplogy measured by the Rocketfuel project [189]. Specifically, we implemented the AS 3967 topology (79 nodes and 147 bidirectional edges), as shown in Figure 2.24, with a single producer (P), single DASH client (C), three adversaries ($Adv1$, $Adv2$, $Adv3$). P hosts a real-time existing MPEG-DASH video ($BigBuckBunny$ movie), both AVC-encoded [126] and SVC-encoded [115]. We separately simulated using three different DASH adaptation strategies: (i) Rate-Based (RB) [116], (ii) Buffer-Based (BB) [187], and (iii) Rate-Buffer-based ($R\&B$) [116]. Other simulation parameters and their values are summarized in Table 2.3.

Following [135, 154], we evaluate the effectiveness of BOA and CoMon-DAS using the following two metrics:

1. Oscillation frequency: The frequency of video quality oscillations within a streaming session.

2. Average oscillation magnitude: The average amplitude of the video quality oscillations within a streaming session.

Table 2.3: Simulation parameters

Parameters	Value
No. of video segments	250
Video period (sec.)	240
Available bitrates (AVC)	20
Layers of quality (SVC)	4
Duration per segment(s)	2
Delay between P to edge router (µs)	200
Consecutive gap α	2
Fragment size s (byte)	1449
MTU	1449
Drop Tail Queue (max. packets)	20
Default cache policy	LRU
Start up delay (sec.)	0.1
Max. buffered time (sec.)	30

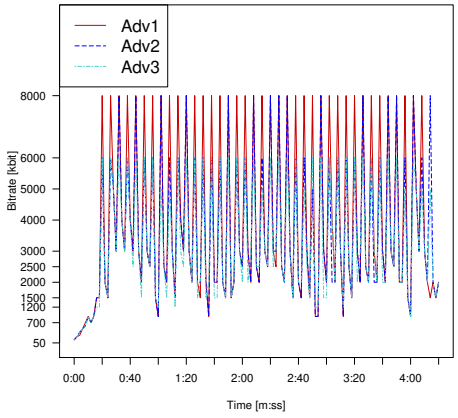

Figure 2.25: DAS RB (AVC) for multiple Adv(s)

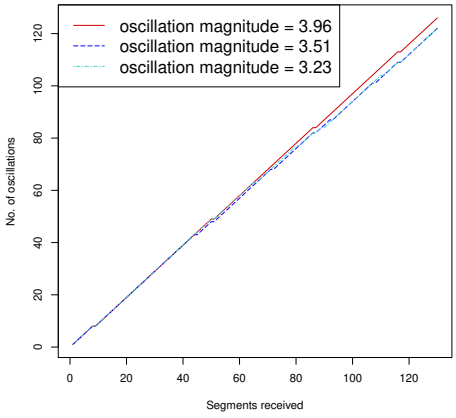

Figure 2.26: Oscillation frequency to various adversarial locations

2.5.4.2 Attack Impact

Figures 2.25 and 2.26 show the adversarial impact of BOA. Specifically, the two figures show an increase in the average oscillation magnitude and the oscillation frequency, respectively, encountered by C for all the cases. These results represent various adversarial locations. Due to space limitations, the rest of the results represent only the case of $Adv1$. However, the results of $Adv2$ and $Adv3$ are very similar, and lead to the same conclusions.

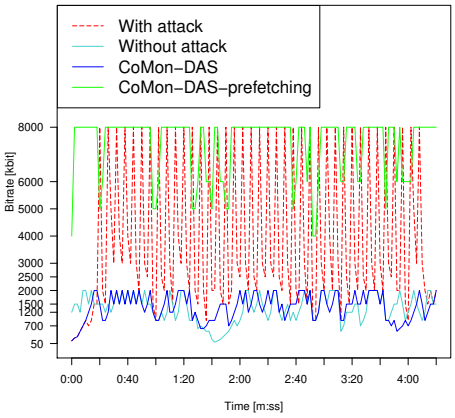

Figure 2.27: DAS applying RB (AVC)

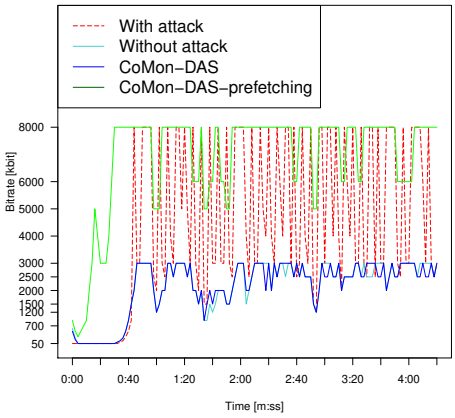

Figure 2.28: DAS applying R&B (AVC)

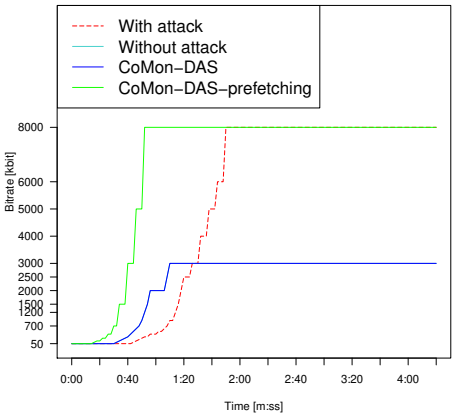

Figure 2.29: DAS applying BB (AVC)

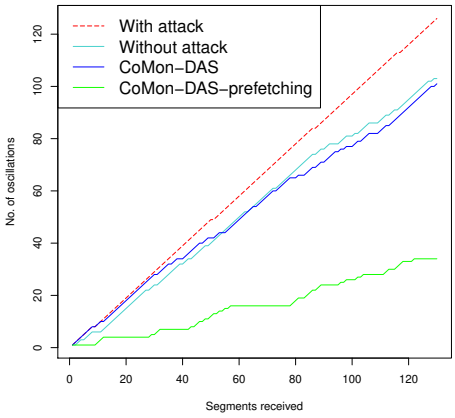

Figure 2.30: Oscillation frequency RB (AVC)

In the case of AVC, Figures 2.27, 2.28, and 2.29 report the bitrate request pattern of the DAS client, with and without BOA, applying RB, R&B, and BB adaptation logic, respectively. The corresponding oscillation frequency results are plotted in Figures 2.30, 2.31, and 2.32. The results show that frequency of bitrate oscillation in the three adaptation logic increases by about 25%. In addition, as can be seen in Figure 2.33, BOA also massively

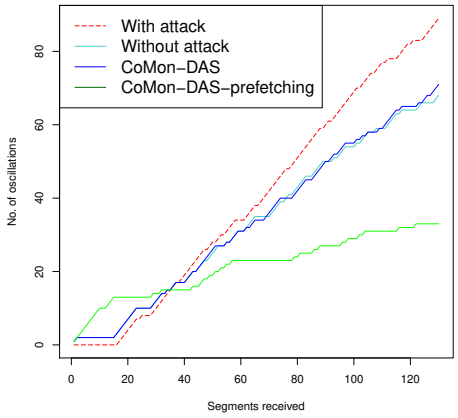

Figure 2.31: Oscillation frequency R&B (AVC)

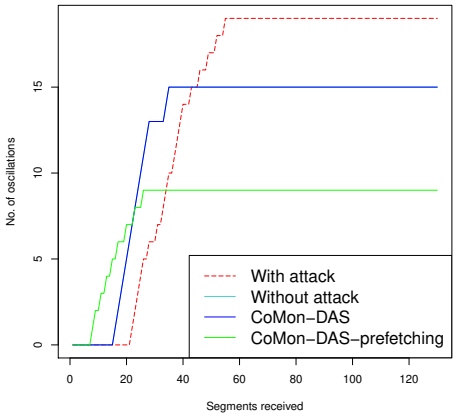

Figure 2.32: Oscillation frequency BB (AVC)

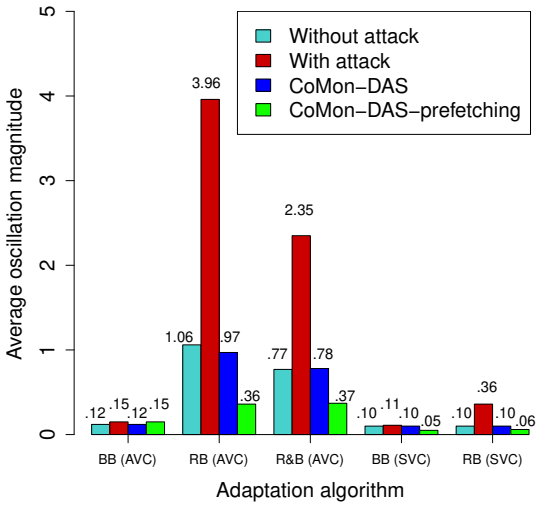

Figure 2.33: Average oscillation magnitude

increases the average oscillation magnitude of bitrate fluctuations. Specifically, the average increases by up to 273%, 157%, and 25%, in RB, R&B and BB, respectively.

In SVC, BOA increases the oscillation frequency for RB, as can be seen in Figures 2.34 and 2.35, by about 275%. Also, Figure 2.33 depicts an

increase in the oscillation magnitude by about 260%, which translates into significant QoE degradation.

The results also reveal that BB in SVC is resilient to BOA (see Figures 2.36 and 2.37). This is because the buffer capacity resists to short term bandwidth fluctuations. However, the use of BB with SVC still remains an open question due to buffer size management in small devices such as smart phones.

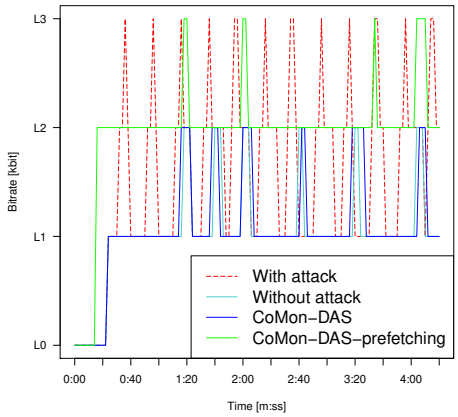

Figure 2.34: DAS applying RB (SVC)

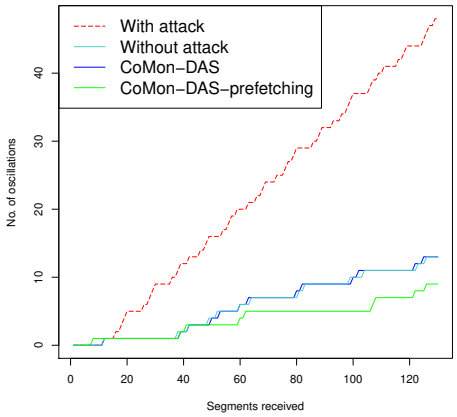

Figure 2.35: Oscillation frequency RB (SVC)

2.5.4.3 CoMoN-DAS Effectiveness

We discuss here the effectiveness of CoMon-DAS with and without dynamic perfecting. Figures 2.27, 2.28, and 2.29 report the performance of CoMon-DAS in presence of BOA, in AVC using RB, R&B, and BB, respectively. The results highlight the phenomena where the victims follow the similar bitrate request pattern as if no attack exists. This is because malicious segments requested by *Adv* are not allowed to cache, thus C's requests follow the original path. Furthermore, the results show that dynamic prefetching of segments provides higher bitrates to C, because sequential segments of higher bitrates are available/retrieved from caches.

Figures 2.30, 2.31, and 2.32 exhibit that the oscillation frequency in AVC with RB, R&B, and BB, respectively, decrease remarkably in presence of BOA. In addition, CoMon-DAS-prefetching enhances QoE by reducing

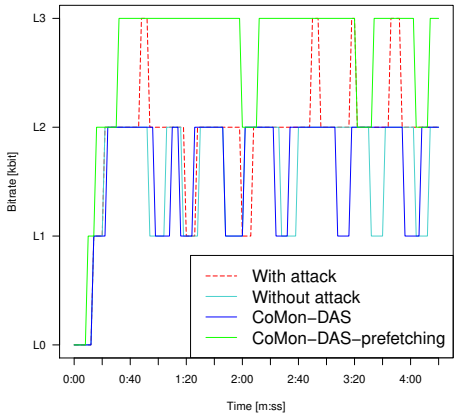

Figure 2.36: DAS applying BB (SVC)

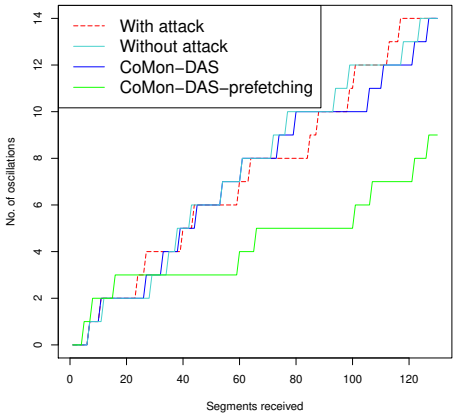

Figure 2.37: Oscillation frequency BB (SVC)

bitrate fluctuations as compared to base-line scenario, by about 70%, 57%, and 46% for RB, R&B, and BB, respectively.

The results depicted in Figure 2.33 confirms the robustness of CoMon-DAS. Specifically, it shows a low average oscillation magnitude in presence of BOA. Also, an additional improvement of 50% to 70% less oscillation magnitude, for RB and R&B, respectively, is achieved by dynamic prefetching.

For SVC, CoMon-DAS results in reduced bitrate oscillations frequency (see Figures 2.34 and 2.35) as well as reduced magnitude (see Figure 2.33) for RB. We can also see that BB in SVC takes advantage of BOA. However, with dynamic prefetching of CoMon-DAS, the client can achieve even better resolution, approximately 50% less bitrate oscillation frequency and magnitude (see Figures 2.36 and 2.37).

All over all, the results report that CoMon-DAS is able to maintain the perceived QoE in presence of BOA. This achievement can be attributed both to (i) the enhanced caching policy (see Algorithm 6) as well as to (ii) the traffic shaping algorithm (see Algorithm 6) which routes the interests requested by the client towards the producer, unless there are sequential segments with optimal bitrates cached in the network. That is, the client would experience higher requested bitrates with reduced oscillation frequency and magnitude.

2.6 Summary

The key features of ICN which includes in-network caching and native tendency to support multicast has been shown to have unforeseen privacy consequences [18]. In this work, first we had shown that how an adversary can exploit the implicit features of ICN (i.e., in-network caching and interest aggregation) and the adaptive streaming mechanism of DASH, to degrade the performance of DASH over ICN. Then, we proposed two countermeasures to detect and mitigate such an attack. Our first countermeasure is a receiver driven approach which uses the concept of maintaining RTT and throughput fairness in ICN's dynamic network condition to alleviate the adverse effects of adversary. Moreover, it shows that it can further enhance the user perceived QoE in presence of varied content source locations and ICN's implicit characteristics. Subsequently, we have also proposed a network based countermeasure, called CoMon-DAS, to protect the network against attack. CoMon-DAS alleviates the effects of adversaries by enabling network-wide coordinated caching and cache-aware routing in ICN. We have extensively simulated the proposed attack and countermeasures in realistic settings. The results show that: (i) high frequency of bitrate switching increases the annoyance factor in spatial dimension, (ii) high amplitude of oscillations decreases the satisfactory visual quality, and (iii) proposed countermeasures can significantly enhance the perceived QoE in presence of varied content source locations and attacks.

Chapter 3

Authentication protocol for ICN based Mobile Networks

Nowadays, the most diffused approach for supporting device mobility is to implement specific mechanisms at link-layer (e.g., tunneling) supported by a dedicated architecture such as Long Term Evolution architecture (LTE). While this approach can handle mobility well within a singular network, it fails to provide a seamless Internet connectivity when mobility occurs among different networks. To achieve inter-networks mobility, researchers proposed to implement mobility management protocols at the network layer. However, the current IP network layer has not been designed for handling mobility, with the result that none of the proposed IP-based methodologies are able to provide a satisfactory solution.

Information Centric Networking is an emerging networking paradigm that gives better support for mobility than IP, enabling complete mobility management at the network layer. In particular, two fundamental ICN design choices facilitate natural support for *consumer mobility*. First, content is addressed by location-independent human-readable names, i.e., they do not reveal any reference to the source or the destination of packets (both interest and content). Next, neither consumers nor producers need a network address (e.g., the IP address) to communicate. Only the name of content is utilised to forward consumers interests towards the corresponding content, and the content back to the requesting consumer. This permits consumers to send interests as soon as an interface is available, as opposed to IP in which a host is bound to wait for a mapping between the interface address and its

layer-3 address. Such content-based, location-independent communication style has been shown to improve device mobility support with respect to the current IP [170], thus raising ICN as a possible future solution to manage mobility at the network layer.

In this chapter, we revise the LTE mobility management and propose a simplified LTE infrastructure that exploits the mobility support provided at the ICN network layer. We revamp the current device authentication protocol for LTE, and we present a novel handover protocol that exploits the ICN communication style. Compared to the protocol adopted in the current LTE, our proposals are able to reduce the number of messages required to authenticate or re-authenticate a device during mobility. We believe that the efficacy of this work is a valid reason to lead network providers for deploying ICN in their cellular infrastructure.

3.1 Authentication and Mobile Management in LTE

The 3GPP consortium [15] defines Long Term Evolution (LTE) and System Architecture Evolution (SAE) to be composed of two main architectural components: the Evolved Terrestial Radio Access Network (E-UTRAN) and the Evolved Packet Core (EPC) network [167]. The LTE architecture is depicted in Figure 3.1.

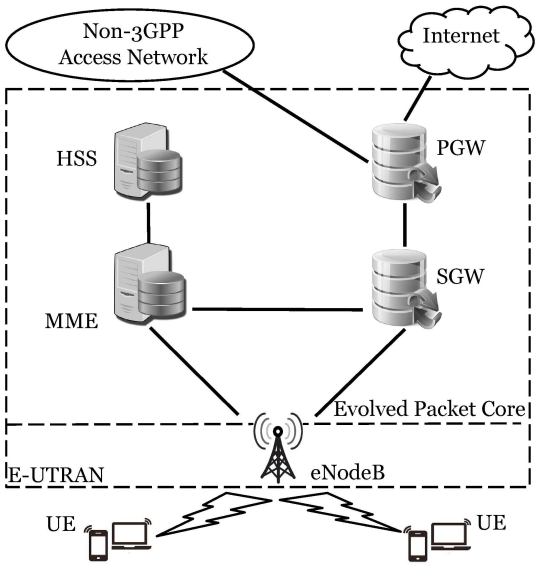

Figure 3.1: LTE Architecture [26]

The E-UTRAN is composed of a number of enhanced node base station, called eNodeB, that provide wireless connectivity to the mobile devices (henceforth called user equipment – UE). The EPC contains different entities used to manage mobility and that compose the SAE. Two notable SAE entities that are of interest for this work are:

- **The Mobile Management Entity (MME).** MME plays a central role for management in LTE/SAE architecture. It contributes mainly in security, authentication, ID allocation of mobile devices (henceforth user equipment - UE) and roaming control in mobility scenarios.
- **The Home Subscriber Server (HSS).** HSS serves as home environment for the whole SAE/LTE architecture containing all the credentials of devices regarding authentication, security, identity and Quality of service (QoS).

Despite these above entities, the core network also have a Serving Gateway (SGW) and a Packet Data Network Gateway (PGW). The role of SGW is to serve the UE by sending and receiving packet data coming from and going to eNodeB, acting also as a limited anchor of mobility service for UE. While PGW connects the core network with other Packet Data Networks such as Internet [41].

In the following, we describe the authentication protocol currently adopted by the LTE infrastructure, the EAP-AKA protocol, and the handover protocol used to manage mobility of the nodes.

3.1.1 Authentication protocol

The authentication protocol adopted in LTE networks is a four-party protocol based on a pre-shared secret key that provides: (i) mutual authentication between UE and the Network, (ii) distributes the necessary cryptographic material to enable ciphering and integrity protection between the UE and the MME, as well as the UE and the eNodeB. The entities involved in the EAP-AKA protocol are:

1. The UE that authenticates to the network.
2. The eNodeB towards which the UE is connecting to the network.
3. The MME which plays the role of an Authentication Center (AuC) that authenticates the UE.
4. The HSS that stores the pre-shared key k with the UE.

The protocol provides mutual authentication between the UE and the network by running the EAP-AKA protocol between the UE and the MME. The HSS will act as an Authorization, Authentication and Accounting (AAA) server, providing to the MME the needed information to perform the EAP-AKA protocol with the UE. Figure 3.2 shows the full authentication phase in the LTE/SAE infrastructure.

The process of mutual authentication starts when UE enters in the radio range of an eNodeB and issues an attach request to the MME. After receiving such request, the MME requests to UE the International Mobile Subscriber Identity ($IMSI$)[1]. Then, the MME requests to the HSS the proper authentication vector (AV) to continue the authentication protocol (i.e., perform the UE authentication and derive further keys to secure the communication with the device). The AV is made of: an token $AUTN$, a random number $RAND$, an expected authentication result RES and a symmetric key K_{ASME}. The triplet $AUTN$, $RAND$ and $XRES$ will be used to mutually authenticate the UE and the network. The key K_{ASME} will be later used by UE and MME to derive further ciphering and integrity keys.

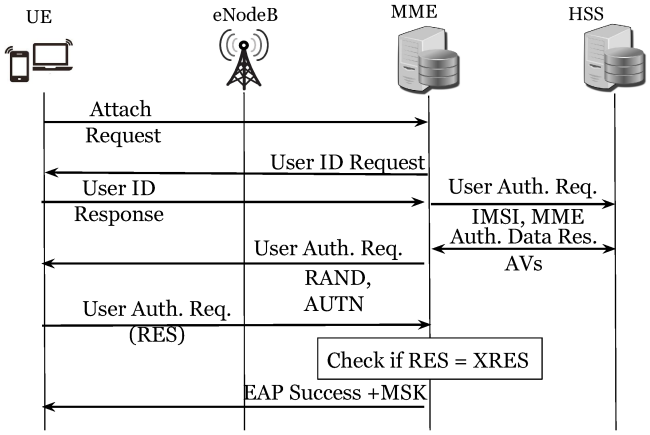

Figure 3.2: EAP-AKA Authentication protocol [26]

Once the MME receives the AV, it issues a user authentication message to the UE passing the value $AUTN$ and $RAND$. Then, the UE derive its own $AUTN$ from the two value k and $RAND$, and it compare its value of $AUTN$ with the value received from the MME. The matches of these two values authenticates the network. The protocol concludes with UE deriving the $XRES$ and forwarding it back to the MME which matches it against

[1] $IMSI$ uniquely identifies a user in a cellular network. $IMSI$ is stored in the Subscriber Identity Module (SIM).

RES. In this case, if the two values matches, the UE is considered to be authentic. Along with $XRES$, the UE derives K_{ASME} too.

Once the mutual authentication is completed, both UE and MME can derive the needed keys to enable ciphering and integrity protection for the communication between the UE and the MME (i.e., Non access stratum security). Moreover, the UE and the eNodeB will both derive ciphering and integrity to protect the message delivery between them (i.e., Access stratum security) [120]. To perform this last step, the MME will share a key K_{eNodeB} with the eNodeB, which will be calculated in the UE too. Such key will be then used to derive the integrity and ciphering keys between UE and eNodeB.

3.1.2 Handover protocol

LTE implements two different handover schemes. The first is a centralized approach in which the MME acts as a connection point receiving the handover requests from the source eNodeB (i.e., the eNodeB the UE is going to leave) and forwarding it to the target eNodeB (i.e., the eNodeB the UE is going to connect to). The second is a distributed approach in which the source eNodeB directly communicate with the target eNodeB exploiting a direct link between them called X2 link.

Each of the two handover approaches goes with its own K_{eNodeB} derivation mechanisms, i.e., every time a UE moves to a different eNodeB, a new key K_{eNodeB} is generated to prevent previous eNodeB (honest or controlled) to decrypt or modify the packet exchanged between the new eNodeB and UE. In the centralized approach, the new K_{eNodeB} is sent from the MME to the target eNodeB, while in the distributed approach the source eNodeB generates and send the new K_{eNodeB} to the target eNodeB.

3.2 Simplified LTE architecture for ICN

In this section, we propose a revised LTE infrastructure in which both the access network and the core network implement the ICN stack. Mobility is managed at network layer in a distributed way as proposed in [32]. Such approach does not require any central entity for managing mobility, such as the MME. For this reason, in our revised LTE infrastructure the MME entity is no longer part of the architecture. The only available entities are:

- **HSS**. Like in the original LTE architecture, HSS contains all the UE information regarding authentication, security, identity and Quality of

service (QoS). In our revised LTE, the HSS is a producer that provides for the content it is storing[2].

- **UE**. A UE represents the device that wants to connect to the cellular network.
- **eNodeB**. The network will be formed of many eNodeBs, acting as point of access to the network for the UEs.
- **ICN Core Router**. In our simplified LTE architecture, the core network is ICN routers.

We assume that all the eNodeBs and ICN routers have the necessary routing and forwarding information to deliver interests to the HSS. Moreover, eNodeBs and ICN routers trust the HSS as the producer for the UE credentials.

This can be achieve either by installing the public key of the HSS in each eNodeB or by involving a root of trust who sign the public key of the HSS, i.e., it creates a certificate for the HSS. In the latter case, an eNodeB has only to verify the HSS certificate once and we assume it to be at bootstrapping time.

3.2.1 Authentication protocol in ICN

We propose an UE authentication protocol in our revised LTE infrastructure that exploits the ICN communication style. Similarly to the LTE authentication protocol, our protocol adopts the EAP-AKA to provide:

- Mutual authentication between the UE and the cellular network.
- Distribution of the cryptographic material to provide integrity and ciphering between UE and the eNodeB connected to the eNodeB.

Our proposal simplifies the original LTE authentication protocol in at least two aspects: (i) it involves three entities (i.e., UE, eNodeB and the HSS) rather than four, thus reducing the communication delay, (ii) it performs the main part of the protocol between the UE and the eNodeB in order to minimize the overall network overhead. While UE and MME are usually multi-hops away one from the each other, UE and eNodeB are instead separated only from one hop. Therefore, running the most of the protocol among UE and eNodeB will reduce the number of messages that

[2]We assume that the HSS publish its content under the namespace */UE/login*.

travels in the network (i.e., from the eNodeB to the HSS). The work in [52] has already shown the advantage of a similar approach. Figure 3.3 depicts our authentication protocol over ICN.

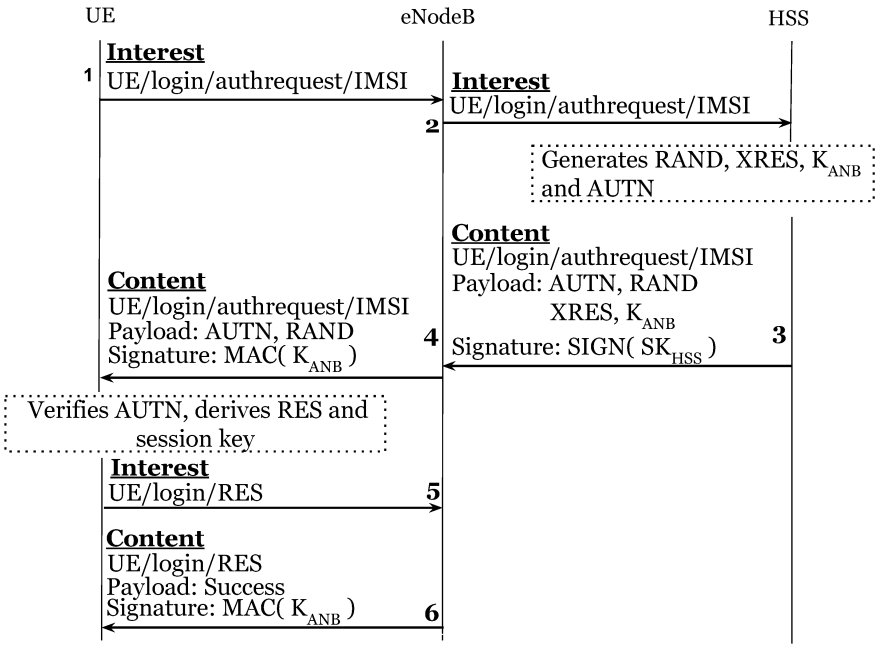

Figure 3.3: EAP-AKA over ICN

The protocol starts with UE issuing an interest requesting to access the network (Step 1). The last component of the interest contains the UE's identity; the $IMSI$. Once eNodeB knows the UE's $IMSI$, it issues an interest to retrieve the AV from the HSS (Step 2). In our proposal, AV is made of: $AUTN$, $RAND$, $XRES$ and a key for access eNodeB K_{ANB}. While the first three are the same parameters used in the original LTE authentication protocol, K_{ANB} is specific for our proposal. Its purpose is allows UE and eNodeB to derive integrity and ciphering key, even in case of handover.

The protocol then continues with the HSS that satisfies the interest issued in Step 2 with a content carrying the AV (Step 3), thus allowing the eNodeB to satisfy the first interest issued by UE with a content carrying $AUTN$, $RAND$ (Step 4). Like in the original LTE authentication protocol, UE calculates its own version of $AUTN$ and match it over the $AUTN$ received from the eNodeB. This check authenticates the network. The protocol then concludes with the eNodeB requesting the authentication from

UE (Step 5), which will reply with a content transporting $XRES$ in its payload (Step 6).

It is worth mentioning that every content packet in ICN must be authenticated with the producer's key (either symmetric or its private key). Therefore, in our protocol we use the HSS's private key, SK_{HSS} to sign each content generated by the HSS, while we use K_{ANB} to authenticate the content exchanged between UE and eNodeB (steps 4 and 6 in Figure 3.3).

3.2.2 Handover protocol in ICN

The handover mechanism in our proposal is performed through an authentication handover module (AHM). AHM is an application running on every eNodeB that prepares relevant eNodeBs (i.e., stores and shares the crypto material to authenticate UE) before UE arrives. In particular, AHM predicts the future location of user [216] and estimates the next area that UE will pass through [3,44,119,165]. Once the area has been calculated, it identifies the group of eNodeBs, namely relevant eNodeBs, covering such area and shares with them the information to authenticate UE.

3.2.2.1 Handover and UE re-authentication

The authentication handover module predicts the set of relevant eNodeBs extrapolating the movement of UE using physical attributes i.e. location, velocity and direction. Moreover, AHM maintains a dataset related to each UE containing its relevant authentication material. Once the relevant eNodeBs have been identified, AHM exploits ChronoSync [227] to share with them the dataset related to the UE.

After the EAP-AKA protocol has been completed, the authenticated UE and eNodeB share K_{ANB}. At this point, AHM starts predicting the set of relevant eNodeBs and it stores and shares a new key K_{ANB*} calculated as follows:

$$K_{ANB*} = KDF(K_{ANB}, RAND). \tag{3.1}$$

When the UE moves to one of the relevant eNodeB, it derives K_{ANB*} and it authenticates to the new eNodeB by sending an interest carrying a new random number $RAND*$ and a message authentication code (MAC) calculated from the interest name (and $RAND*$, later used again) with K_{ANB*}. The eNodeB then replies with a content authenticated with K_{ANB*}. If both interest and content are authentic, then UE and the eNodeB are authenticated by each other and they can further derive ciphering and integrity key for securing their communication. At this point, AHM running in the eN-

odeB starts predicting the set of relevant eNodeBs and it shares a new key K_{ANB*} calculated as described in Equation (3.1) (in this case, the previous value of K_{ANB*} will replace K_{ANB} and $RAND*$ will replace $RAND$ in the equation).

3.2.2.2 Synchronization of the key access eNodeB in the relevant eNodeBs

Synchronization of K_{ANB*} is performed through the ChronoSync protocol. ChronoSync synchronizes the state of a given dataset among multiple ICN entities [227]. The protocol works on the idea to encode the state of the dataset of each entity into crypto digest form (i.e., SHA256) called *statedigest*, or digest in short. These state digests are then exchanged among all the entities participating in particular synchronization group. Each entity depending upon the state of its own dataset calculates the state digest, and sends a broadcast interest to all the other entities in that group, containing that state digest. On receiving such interest, if the value of the incoming digest is identical comparing to the value maintained locally, no action will be taken and called as stable state. Otherwise, the difference of the dataset state is directly inferred and sent in response to the sync interest [227]. With the knowledge of the up-to-date state dataset, an ICN entity (or one of its running application) can then decide to fetch the new content in the dataset.

In our proposal, ChronoSync synchronizes the state of UE's dataset, i.e., the key K_{ANB*}, on each relevant eNodeB predicted by the AHM. Therefore, the AHM's dataset running on each relevant eNodeB will be notified of the new key K_{ANB*} and will fetch it from the eNodeB sending the notification. After the new K_{ANB*} is fetched, a relevant eNodeB is ready to authenticate UE and to perform the handover as explained in Section 3.2.2.1. Figure 3.4 shows the communication between relevant eNodeBs during the synchronization process. The synchronization of the dataset state and the fetching of K_{ANB*} require the definition of two namespaces, namely the `sync data namespace` and the `application data namespace`. The `sync data namespace` is used to carry interests and contents used to synchronize the dataset state using ChronoSync. The purpose of `application data namespace` is to have routable name prefixes, so that interests can be forwarded towards the relevant eNodeBs directly, as AHM behaves like a producer in each eNodeB. Figure 3.5 and 3.6 show an example of content name in the `sync data namespace` and in the `application data namespace` respectively. Figure 3.5 shows a content in the application data namespace.

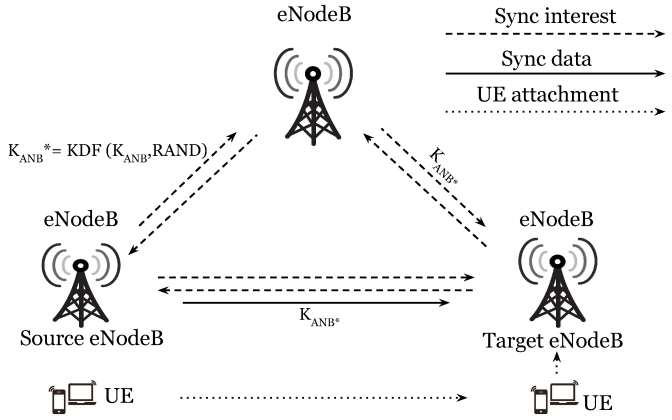

Figure 3.4: Synchronization

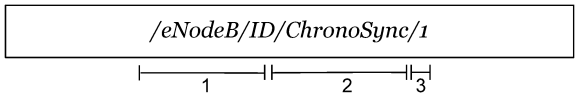

Figure 3.5: Application data name

The first part of the application data name (indicated as 1) represents the routable prefix for particular eNodeB with its unique ID. Part (2) represents the name of a particular application to synchronized. It shows the name of the process which is responsible for handling that particular interests. The data generated by eNodeB is named sequentially, for example with the initial value of $K_{ANB}*$ computed by AHM has a sequence number zero. Whenever a new value of $K_{ANB}*$ is generated, this sequence number is incremented by one. So, the last part (3) is the sequence number of the latest $K_{ANB}*$.

Sync data namespace, depicted in Figure 3.6, also consists of three parts. Part (1) is the prefix ensuring the broadcast namespace for the given domain created by AHM ascending index. In particular, such prefix will be shared among all the eNodeBs along propagating path of the user (index i.e., eNodeB1, eNodeB2, eNodeB3, ...). This will allow a synchronization interest to reach all the relevant eNodeB. In the Part (2) similarly as application data names defines the name of application, which shows that particular interest are responsible for authentication request of specific user. The last part notifies the recent state digest of the interest sender, i.e., the digest of its current $K_{ANB}*$.

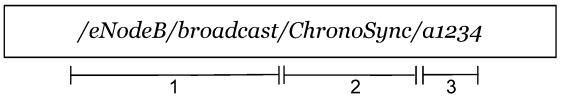

Figure 3.6: Sync data name

3.3 Evaluation

In this section we compare the performance of UE authentication in our proposed LTE infrastructure for ICN with today's LTE infrastructure. The performance comparison evaluates the delay occurred during authentication of a UE, and also re-authentication during handover comparing to handover authentication in LTE, particularly in the distributed handover.

3.3.1 Authentication delay evaluation

In order to evaluate the authentication delay required in the two infrastructure we define the time taken by the method to complete the authentication process as the total authentication delay (D_{auth}). D_{auth} can be further divided into three components: the delay of the EAP messages transmission (D_{trans}), the EAP messages treatment delay (D_{tre}) considering data base access, key and tag generation, computation, encryption/decryption, and the propagation delay (D_{prop}) [100].

D_{tre} is the delay occurred during EAP messages treatment/processing on each node, which depends on LTE servers and UE performance (e.g., CPU, memory). We assume that our proposed protocol and standard EAP-AKA use same key encryption with similar key sizes. Therefore we can say that treatment delay is identical in both protocols, and also considering performance of LTE servers, we assume transmission delay is insignificant. Thus total authentication delay depends upon the propagation delay D_{prop} [100].

D_{prop} can be divided in four sets: $D_{prop(UE-eNodeB)}$ propagation delay between UE and eNodeB, $D_{prop(eNodeB-MME)}$ propagation delay between eNodeB and MME, and $D_{prop(MME-HSS)}$ propagation delay between MME and HSS. The total authentication delay for EAP-AKA in the current LTE then can be expressed as [100, 185]:

$$\begin{aligned} D_{auth(EAP-AKA)} = D_{tre(EAP-AKA)} + 5D_{prop(UE-eNodeB)} \\ +5D_{prop(eNodeB-MME)} + 2D_{prop(MME-HSS)}. \end{aligned} \tag{3.2}$$

Table 3.1: Authentication delay comparison

Authentication Protocol	Authentication delay ms
EAP-AKA	1244
ICN(EAP-AKA)	1019

From Figure 3.2 and Equation (3.2) we calculated that total number of messages exchanged between entities i.e. UE, eNodeB and MME (which are 5, 5 and 2 respectively) multiplies the propagation delay between them. Also, the total authentication delay of EAP-AKA in our ICN based architecture can be expressed as, Figure 3.3:

$$D_{auth-ICN(EAP-AKA)} = D_{tre(EAP-AKA)} + 4D_{prop(UE-eNodeB)} + 2D_{prop(eNodeB-HSS).} \tag{3.3}$$

From equations (3.2) and (3.3), we note that our approach reduces the propagation delay between UE and eNodeB by one. This is because EAP-AKA over ICN requires one message less that the EAP-AKA protocol adopted in the current LTE.

Another improvement in the authentication delay is due the removal of the MME in our approach. This, in turn, reduces the number of messages exchanged with the UE and entities in the EPC. In particular, our approach exchanged only two messages between the UE and the HSS, with a propagation delay that is indicated as $2D_{prop(eNodeB-HSS)}$ in the Equation 3.3. Instead, the current LTE requires 5 messages exchanged from the UE and the MME, and 2 messages exchanged from MME and HSS. The propagation of those messages is indicated in Equation 3.2 as $5D_{prop(eNodeB-MME)}$ and $2D_{prop(MME-HSS)}$, respectively.

Table 3.1 compares the authentication delay of the EAP-AKA in LTE and our EAP-AKA over ICN. We assume that $D_{prop(eNodeB-HSS)} \cong D_{prop(eNodeB-MME)} + D_{prop(MME-HSS)}$ which we consider to be a pessimistic assumption for our approach. We expect that an architecture without MME will not increase the propagation delay between the UE and the HSS, but in the best case it will reduce it. However, since we cannot evaluate such improvement, we compare the authentication delay under the worst case scenario for our mechanism.

To evaluate the authentication delay we used the experimental values found in [100] and [186]. In [100] the average value for complete EAP-AKA authentication delay is 1244 ms. In [186] authors approximated the propagation delay between eNodeB to MME is 75 ms. We used such value

Table 3.2: Handover Authentication delay comparison

Handover Authentication Protocol	Authentication delay ms
EAP-AKA	34.30
ICN(EAP-AKA)	20.58

to calculate the average authentication delay in our protocol, which results 1019 ms.

3.3.2 Handover authentication delay

While calculating re-authentication delay using same Equation (3.3), we found that it depends on the propagation delay between source to target eNodeBs. The handover scenario we have assumed during evaluation for LTE is the X2-based handover. In X2 handover, authentication material derived during full EAP-AKA is transferred by source eNodeB directly to target eNodeB exploiting the direct X2 link. This scenario, also named as horizontal handover is fair to compare with our proposed protocol, as we also proposed a network infrastructure without the MME. Therefore for LTE re-authentication delay during inter eNodeB/X2 handover can be calculated as the number of messages exchanged from source to target eNodeBs.

$$D_{hand-auth(EAP-AKA)} = D_{prop(Src_{eNodeB}-Trg_{eNodeB})}. \tag{3.4}$$

From the work in [140], we calculate the total number of messages between eNodeBs and therefore:

$$D_{hand-auth(EAP-AKA)} = 5D_{prop(Src_{eNodeB}-Trg_{eNodeB})}. \tag{3.5}$$

Also using same equation for calculating handover authentication in our proposed protocol Figure 3.4, we found:

$$D_{hand-auth-ICN(EAP-AKA)} = 3D_{prop(Src_{eNodeB}-Trg_{eNodeB})}. \tag{3.6}$$

From equations (3.5) and (3.6), we evaluated that our re-authentication protocol is requires two less messages propagated between eNodeBs than the handover in LTE. We expect that decreasing the number of messages will produce a lower handover authentication delay. To confirm it, we estimate the two handover authentication delays and we report our comparison in Table 3.2.

To evaluate the two handover authentication delay, we estimated the propagation delay from source eNodeB to target eNodeB on X2 link.

From [212], we calculated that the propagation delay is approximately 6.86 ms. Using equations (3.5) and (3.6) we evaluate that our handover authentication protocol reduces the authentication delay by the value of approximately 13.72 ms.

3.4 Security Analysis

In this section we provide a security discussion about the two protocols that we propose: the authentication protocol and the handover protocol.

3.4.1 Authentication protocol

We argue that the security of our authentication protocol is comparable to the security provided by the authentication protocol in LTE. Both the authentication protocols exploit the standard EAP-AKA protocol without changing any of the steps described in the EAP-AKA specification. Therefore, all the security considerations made for EAP-AKA are still valid for our EAP-AKA over ICN [30].

3.4.2 Handover protocol

Our handover protocol is resilient to an external adversary (i.e., an adversary that does not own a valid $IMSI$ for the HSS) with the goal of authenticating itself to an eNodeB. In order to achieve it goal, the adversary must be able to issue an interest with a valid MAC to the eNodeB. However, to be able to successfully generate the MAC for the interest the adversary must know, or obtain, a valid $K_{ANB}*$.

Our handover authentication protocol make it unfeasible for an external adversary to obtain a valid $K_{ANB}*$. This due to the fact that $K_{ANB}*$ is never transmitted between UE and the target eNodeB, but rather calculated from K_{ANB} and $RAND$. Only $RAND$ is transmitted between a UE and the target eNodeB, therefore as long as the key derivation function requires only K_{ANB} to be secret, the adversary has no way to derive $K_{ANB}*$. Unfortunately, the adversary might eavesdrop and replay $RAND$ to another eNodeB and authenticate to it as a genuine UE. While this attach will not let the adversary to communicate through the network (it will not be able to generate the correct integrity and ciphering keys, which are derived from K_{ANB}), it can temporarily waste some state in the eNodeB. In fact, every time an eNodeB authenticate and UE (genuine or not), it must reserve the necessary state to handle a communication with it (e.g., the integrity and

ciphering key). To protect the network from such replay attack, we propose to adopt a physical-layer authentication.

Physical-layer authentication uses the subtle features of the physical-layer signal to provide a secure device authentication between two trusted nodes in the presence of an adversary/eavesdropper with unlimited computational power [144, 215]. We exploit the RF finger printing technique [197] to provide physical layer authentication in our handover authentication protocol. The RF finger print mainly depends on the differences of each trusted transmitter components, power supplies and environmental factors, which are extracted from RF signal. From [144] different equipment/nodes (UEs) can be identified by measuring the specific value extracted from their unique RF fingerprints denoted as $|A(t)|$. In our handover authentication protocol we use $|A(t)|$ as an entity replacing $RAND$ to provide physical-layer authentication. So in Equation (3.7), K_{ANB*} is the key derivation function of key for source eNodeB and RF finger print of the UE authenticated initially to mitigate the presence of eavesdropper. However, the functioning of AHM for predicting the relevant eNodeBs and ChronoSync protocol to synchronize the dataset among multiple ICN entities will follow the same as described above.

$$K_{ANB*} = KDF(K_{ANB}, |A(t)|). \tag{3.7}$$

3.5 Summary

In this chapter, we propose a revised LTE infrastructure that exploits the ICN communication paradigm to manage UE authentication and transporting the UE security context from the old eNodeB to the new one. We design a new handover mechanisms that does not require any central entity, e.g., the MME, to distribute the cryptographic material to the new eNodeB.

Our approach reduces the complexity of the LTE infrastructure thus making it simpler, easier to manage and more cost-effective for network providers. We believe that this is a valid reason that would lead network provides for deploying ICN in their cellular infrastructure.

Chapter 4

Secure Mobility Management in ICN

Information Centric Networking has gained significant attention from both academia and industry as it satisfies the fundamental requirements (e.g., mobility management, security and efficient content distribution) of next-generation heterogeneous mobile networks. Along with security and in-network storage, the result of decoupling time and space among request resolution and content transfer enables ICN to provide seamless mobility as an instinctive characteristic of the network architecture. In ICN, the consumer mobility is supported by design in virtue of its connectionless pull-based communication model. However, producer mobility focuses on the named-based resolution mechanism, which applies a dynamic and direct interaction between the producer and forwarding plane.

In this chapter, we consider the fundamental security issues related to the *producer mobility* in ICN, i.e., insecure interaction of producer with the network forwarding information management system. We identify that installing such protocols which are deprived of acceptable security mechanisms bring up serious security threats for all network entities, e.g., consumer, producer, and the network itself. In this regard, the producer should be allowed to issue only the legitimate routing updates, explicitly named as Interest Updates (IUs), for the prefix(es) that it is entitled to publish the relevant content. In cases where no adequate security mechanism exists to impose such rules then an adversary is able to easily forge IUs of the legitimate producers. Hence, it can divert benign consumers request and network traf-

fic towards itself, such attack in ICN is known as prefix hijacking [33]. By launching prefix hijacking, an adversary is able to: (i) victimize benign users by performing blackhole attack [25], (ii) deny consumer's access to their requested content [89], (iii) make genuine content reachability unavailable, and (iv) pollute the network caches with false content [58].

Currently, BlockChain (BC) is gaining significant attention from both academia and research industry where researchers exploit BC technologies to assure security, privacy, and access control for devices, data storage, and various other applications [69,96,145]. Driven by the importance of addressing security issues in the initial stages of potentially new Internet architecture (i.e., ICN), we propose a BC based efficient & lightweight distributed mobile producer Authentication (BlockA*uth*) protocol for mobility management in ICN. BlockA*uth* authenticates the producer prefix(es) to enforce them to express only genuine IUs. Our qualitative security analysis confirms that BlockA*uth* is robust against various security attacks to which mobile network and blockchain are particularly vulnerable (e.g., prefix hijacking, double spending, DoS attack). In addition, the performance evaluation of BlockA*uth* shows that it maintains significant performance gain compared to the state-of-the-art prefix attestation proposals.

4.1 Mobility management in ICN

In contrast to IP networks where handling mobility requires cumbersome solutions such as Mobile IP [63, 163], ICN provides native mobility support to consumers. Two fundamental characteristics of ICN architecture supports seamless consumer mobility [28, 76]. Firstly, the communication model is the receiver(consumer)-driven instead of producer, where the consumer uses the location-independent content names to request data, while in current Internet architecture sender has complete control on data transfer. Secondly, the request/response communication model of ICN between consumer and producer is connectionless (i.e., *stateless*). It is in contrast to current TCP/IP connection-oriented (*stateful*) end-to-end communication which requires a binding between user location and address. Therefore, when mobile consumer attaches to a new Point of Attachment (PoA), the two above-mentioned characteristics permits the consumer to reissue the interests to obtain the data, which he/she did not received at its previous PoA. In this way, consumer achieves seamless mobility support in ICN, deprived of rebuilding a TCP connection or by means of cumbersome and overwhelm IP mobility patches [163]. However, the producer mobility is more challenging in ICN because of no separation between routing locator and the content

identifier. In particular, for each mobility event initiated by a producer, the network should maintain producer reachability, and the routing devices must adjust their forwarding information so that the interest(s) matching the prefix(es) owned by producer can be re-directed to its new location. Thus, unlike consumer mobility, ICN requires updating of name resolution system over the new location of the producer to maintain routing consistency during content provider attachment to a new PoA [28, 76].

In past, few proposals for handling producer mobility are proposed [28, 223]. The solutions like *indirection-based* and *resolution-based* supports producer mobility, however, these also bring complexities to few intrinsic problems such as handoff latency and packet overhead during encapsulation and decapsulation that leads to QoS degradation [109]. The *routing(tracing)-based* approaches try to address the subject by updating the forwarding table at each mobility event. In particular, the tracing-based protocols [32, 95, 109, 206, 224] directly exploits the ICN stateful forwarding plane to overcome the handoff latency, packet loss, and signalling overhead. However, the tracing-based protocols allow the producer to directly interact with the network forwarding information. Hence, installing such protocols deprived of acceptable security mechanisms could cause serious security threats for all the ICN entities.

4.2 State of the Art Security Issues and Related Work

In the literature of IP based mobile networks, the prefix hijacking attack is mitigated by adopting prefix attestation mechanisms in IP-based mobility protocols such as Mobile IP [163], Cellular IP [40], and TeleMIP [62]. In all these approaches, the host has been assigned with a host ID and session key by the network gateway during its initial attachment to the gateway. In case of handover, the host uses the session key for the purpose of IP address authentication owned by it to the new PoA. However, the approach brings some limitations when it comes to ICN-based future mobile networks. Since the above mentioned protocols follows a centralized authentication mechanism, therefore, whenever the host changes PoA, a central entity is needed to authenticate the host [163]. It is worth noticing that the efficacy of ICN tracing-based mobility protocols relies on the elimination of any central entity managing the mobility of producers to overcome handoff latency. In other approaches such as [40, 62], the use of a single network key to generate host's session keys is also problematic. For instance, if the network key

is compromised, then any router can be compromised and new network keys along with the regeneration of all sessions keys is needed to tackle the issue. In addition, these mechanisms do not provide any solution to identify the malicious network routers and malicious legitimate users which tries to perform repudiation or replay attacks.

Some authors also addressed the issue of IP prefix hijacking in inter-domain [12, 98] and intra-domain [149, 204] IP routing. The authors propose prefix attestation to mitigate the threat. One common mechanism to achieve IP prefix attestation is by exploiting digital signatures and certificates, in which an address owner requests a signed certificate to attest the routers right to announce specific IP addresses in the network. For instance, sBGP [105] and soBGP [208] make use of public key infrastructure to maintain trust between the network and the address owners. Correspondingly, authors in [149] propose the use of signed certificates for the network prefixes which OSPF routers need to announce in different OSPF areas. The similar approaches can be applied to the ICN tracing-based mobility protocols, however, it would suffer with the similar issues.

Recently, authors in [53] highlight the issue of prefix hijacking [33] in ICN. To address it, authors propose a distributed prefix authentication protocol for producer mobility in ICN routing-based protocols. The protocol utilizes a one-way hash-chain mechanism to guarantee that a producer can only generate IUs for its own prefix(es). Moreover, to achieve the forward secrecy, the network and the producers are always forced to maintain a synchronized state of the current hash chain value, which is used for prefix authentication. We identify that similar to other approaches, the protocol in [53] also do not completely prevent the prefix hijacking attacks. For instance, there is no guarantee that every router in the network will have the most recent version of the forwarding state, i.e., a synchronized sequence number of the current hash chain value. Hence, the outdated routers could not be able to detect old or replayed IUs if the IUs holds a greater sequence number than the security context stored at the router. Secondly, the one-way hash computation also encourages DoS attacks to the edge routers. For instance, adversary issues a non-legitimate IU holding a sequence number (say j), such that $j >>> i$, where i is the recent sequence number that have been used at the edge router. As a result, to detect the non-legitimate IU, the router is forced to hash $j - i$ times the security context associated with the prefix. The greater is the distance between j and i, more the number of hashes router have to compute. An attacker can issue non-legitimate IUs with great sequence numbers to keep the router busy on calculating such hashes, thus it provokes a DoS attack to the other connected producers.

Additionally, the protocol also lacks to mitigate the double spending attack, e.g., repudiation by a legitimate producer. Categorically, it is the case where a legitimate producer tries to use the same sequence number for a hash value (nearly at the same time) on two different edge routers in the network. The producer can re-route legitimate traffic to or for that producer on different paths since the hash chain is not able to identify the double spending of sequence number by the producer. Finally, the protocol also lacks to provide the mechanism to identify malicious routers and producers which are connected to the network.

4.3 Blockchain

BlockChain (BC) [113] is an immutable time-stamp ledger of hashed blocks which functions to store and share data in distributed manner [210]. In recent years, practitioner and academics in diverse disciplines (e.g., law, finance, and computer science) have been tremendously attracted by BC due to its noticeable features which includes distributed structure, immutability, security and pseudo-anonymity [37, 69, 71, 80]. More specifically, the basic structure of blockchain is similar to linked list data structure. Each data block is hashed and linked with the previous block to provide immutability. The data block contains historical, verified instances (named as transactions in Bitcoin), information, and control data. A chain of blocks can be replicated and spread to all participants in the BC network such that the data contained in the BC is synchronized globally. Figure 4.1 illustrates the basic structure of BC.

Each new transaction is verified and confirmed by all the participating nodes in the network, therefore, it eliminates the necessity for any central authority. Appending a new block to the BC (referred to as a mining process in literature) may entails solving a computationally demanding, hard-to-solve, and easy-to-verify puzzle. The puzzle indeed supports a trust-full consensus algorithm among the untrusted participating nodes. Typical consensus algorithms used in blockchain implementation are Proof of Work (PoW) [203] and Proof of Stake (PoS) [210]. To propagate the transactions and blocks to update the ledger, BC operates with multi-hop broadcast functionality.

BC [113] technology was first devoted to power the bitcoin cryptocurrency, but nowadays it is progressively proving its applicability to various other applications. For instance, authors in [229] utilizes BC in order to guarantee privacy aware and secure personal data management. In [82] authors describe decentralized secure content access using BC in ICN based platform. Similarly, numerous authors shows the usability of BC to provide

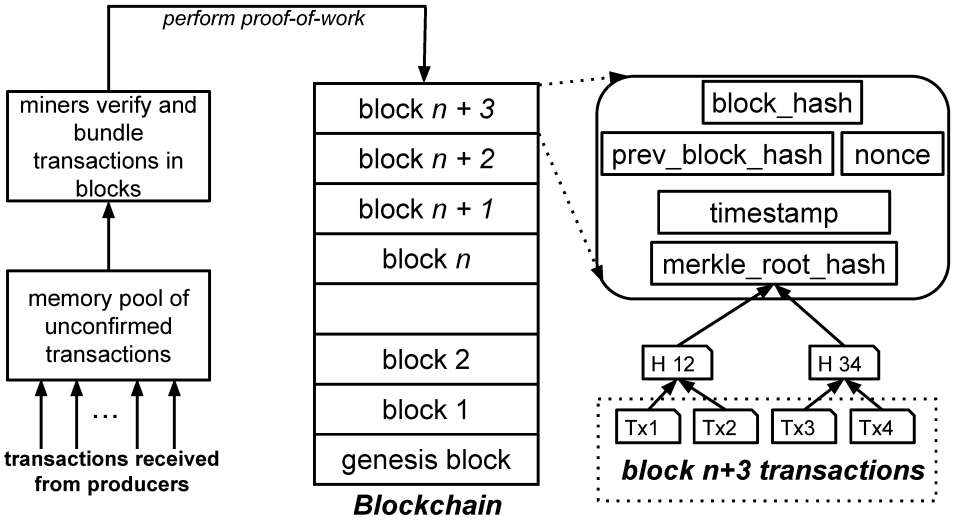

Figure 4.1: Blockchain structure

the authentication to nodes and create security and trust in Wireless Sensor Networks (WSN) [145] and Internet of Things (IoT) [17, 69, 71, 72]. In our proposed protocol, we utilize BC to store the access control data in a decentralized manner to authenticate mobile producer interaction with network forwarding information. In addition, with the use of BC we assure trust and security in the network in a distributed manner.

4.4 BlockA*uth*: BlockChain based Distributed Producer Authentication

In this section, we present the design and working methodology of our proposed BC based efficient & lightweight distributed mobile producer Authentication (BlockA*uth*) framework, which offers a reliable and fast mobile user authentication. It also mitigates various security and privacy issues of mobility management in ICN.

4.4.1 BlockA*uth* Framework Overview

To perform efficient handover in inter and intra-clusters [94,121,133,192], the BlockA*uth* framework consist of two main tiers namely: core network, and clusters[1] (i.e., micro cells). To perform the relevant operations, the clusters may change dynamically, in fact, reflecting the mobility of the underlying

[1]The concept of dividing the geographical region into small zones has been presented essentially in the literature as clustering.

network. Each cluster consists of a group of access gateways (e.g., LTE base stations, and WiFi access points) and one cluster head. In BlockA*uth*, some of the ICN routers which entails the necessary functionalities such as processing capability and bandwidth availability are selected as cluster-head (CH). The base stations register with the nearest CH to become the member of the respective cluster. BlockA*uth* utilizes the *weighted clustering algorithm (WCA)* [45] which takes into consideration the number of base stations that a cluster-head (i.e., edge router of the core network) can handle efficiently without any severe degradation of the system performance. In particular, while selecting CHs, WCA considers the transmission power, mobility, and battery power of the mobile nodes.

To ensure fast and seamless intra (also named as micro) cell handover to reduce signalling overhead, the base station (or access point) within a cluster uses a private expandable Immutable Ledger (IL) entailing the transactions called as local transactions. Categorically, the structure of private IL is similar to BC, however, it is managed solely and centrally by the respective CH, later we name it as Local Immutable Ledger (LIL). To guarantee the requirements of handover latency and scalability, only the CH is responsible for managing private LIL. The CH is thus also named as Local BC Administrator (LBA).

In BlockA*uth*, the capable routers in the core network collaboratively manages the private Global BC (GBC), which stores the transactions generated to and from various clusters to perform inter cluster (or macro) handover [94]. The BC in the core network is managed by a subset of the core routers which we call Global BC Administrators (GBA). Thus, the routers participating in Global BC (including CHs) are responsible to process incoming and outgoing transactions that are generated by mobile producers from different clusters. Note that the routers selected as GBAs and LBAs (i.e., CHs) are expected to have sufficient resources to process the generated blocks and transactions.

4.4.2 System And Adversary Model

In this chapter, we consider the scenario of anchor-less forwarding [32, 225] for mobility management which is considered as a promising requirement of the forthcoming next-generation mobile networks. Our proposed model follows the scenario of the handover mechanism between base stations similar to [16, 68], and it includes macro and micro mobility scenarios as illustrated in Figure 4.2. However, it does not specialize to any specific wireless media, i.e., the access gateways can be LTE/4G/5G cell or WiFi access points.

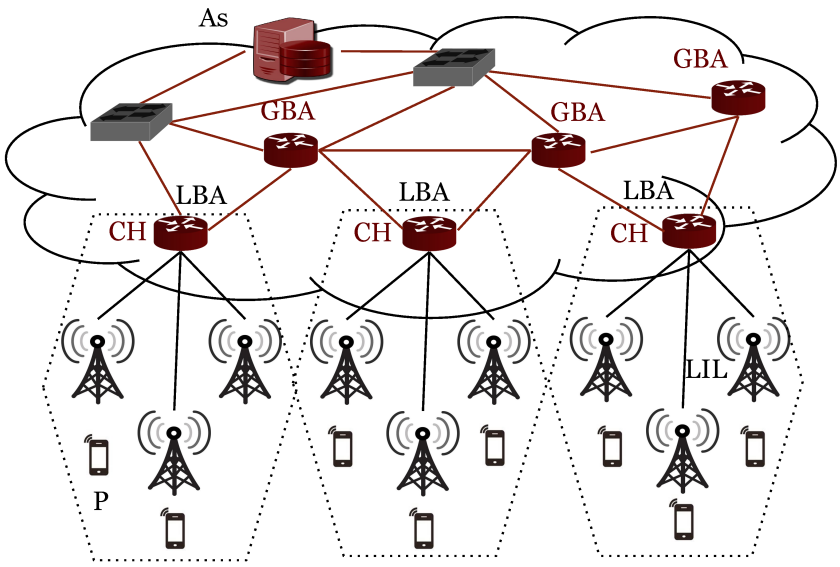

Figure 4.2: System model

The major entities involved in our protocol are Authorization Server (As), core routers, and cluster heads. The CHs are responsible to manage their respective cluster members and mobile producers attached to them. The subset of core routers functioning as GBA manages the GBC, and if the GBA is CH than it manages the LIL as well. Furthermore, we consider that the composition of the network containing the edge and core routers forms a single autonomous system. The mobile devices are aware of their valid credentials to connect to the relevant network infrastructure, i.e,, producers are provided with valid SIM card devices by the network operator. Moreover, once the authentication is performed, the communication between each mobile device and access gateway is considered secure. Table 4.1 summarize the notations used in our protocol.

4.4.2.1 System model

To accomplish the verification of IUs that are issued by a mobile producer (P), we implicitly categorize P as a mobile device. The device stores $P's$ identity which is entitled to publish content using one or multiple $prefix$(es). Each P is associated with a pair of public/private key, say pk_P^s and sk_P^s respectively, which are used to sign/verify the $prefix$(es) it publishes. We assume an additional field attached to IU along with the prefix, which entails the security context to verify the same prefix.

Table 4.1: Summary of notations

Notation	Meaning
As	authorization server
BA	blockchain administrator
GBA	global BC administrator
LBA	local IL administrator
R_i	set of routers
Bs	base station/access gateway
CH	cluster head
P	mobile producer
Adv	adversary
IU	interest update message
$prefix$	producer prefix
$H(.)$	cryptographic hash function
pk_P^s, sk_P^s	public and private signing key pair associated with P
$enc(.,.), dec(.,.)$	public key encryption and decryption function
Tx^{ID}	transaction ID
$previous\_Tx^{ID}$	previous transaction ID
Tx^{Max}	maximum transaction a block consists
Tx_i^{ID}	local IL transaction ID
Tx_j^{ID}	global BC transaction ID

In our protocol, authentication server (As) is mainly responsible to perform the:

- authentication of P and verification of the $prefix$(es) owned by P,
- generation of the genesis transaction for BC which serves as a starting point of the global BC.

4.4.2.2 Adversary Model

We consider an adversarial model in which an adversary (Adv) is capable to control mobile devices that can attach to the network, e.g., attacker owns valid SIM cards and connects to the network. The Adv could target the authorized mobile producers and purposely generate legitimate IUs for the $prefix$(es) used/owned by its victims. In addition, a legitimate P can also be an Adv, and it aims to corrupt the network by double spending attacks. For instance, P can try to use the valid security context for the $prefix$ authentication on two or more different base stations (Bs) at the same time. We also assume that Bs, CH, and core routers can be compromised

by the Adv, while the As is considered a trusted entity. These assumptions are consistent with the assumptions made on the existing heterogeneous mobile networks [14]. Moreover, we assume there is no intrusion detection mechanism in place. Finally, we also assume that Adv can access information stored at the compromised ICN router (R_i) including Bs, LBA, and GBA nodes.

4.4.3 Modelling Data through BlockChain

As previously illustrated, BC functions as a transaction database which is distributed and shared among all nodes participating in the BC network. We exploited BC as an application for distributed data storage which provides various functionalities for data storage [72, 96]. Among them, three fundamental primitives that are essential to our proposed solution are: (i) retrieve, (ii) update, and (iii) add. In this section, we illustrate how BC provisions these primitives and enables the data flow in our proposed model. However, first we review some key definitions of BC characteristics in the light of BC adopted by Nakamoto [151], as illustrated in Figure 4.1.

- Transaction: It is the value to transfer (e.g., information, currency). In our case, similar to the Bitcoin transactions, it always devotes all input values to the new outputs, and it will be done as a chained process by referring each previous transaction outputs to new transaction inputs. It is possible to fetch and read each transaction that is ever placed in the blockchain. To be precise, in our method we express each $P's$ $prefix$ authentication request (i.e., IU) as a single transaction in BC.

- Block: Multiple valid transactions are first clubbed together to form a block, and then the blocks are verified before storing into the blockchain. In addition, BC shapes the blocks linearly in a chronological manner over time. Same as the transactions, each block also delivers immutability by containing the hash of the previous block. The new transactions are being processed and added into a new block at the end of the BC in such way that a transaction can never be altered or removed once being added into the BC.

- Mining: It is a process to provide distributed consensus in the system. Mining validates each transaction, create blocks, and then verify the blocks to add them in the blockchain. The block addition process impose a chronological order in the BC, and it provides a state of agreement among all participating nodes. The nodes performing the

mining are known as miners. In BlockA*uth*, the process of mining is only performed by BC administrators, i.e., GBAs and LBAs. Therefore, only a subset of core routers and CHs are responsible to mine and broadcast the new blocks into the network.

- Genesis block: It is the foremost block in the blockchain. In the typical scenario, the genesis block could be hard-coded into the software, and it is a special case in which it does not refer to any previous block in BC. In BlockA*uth*, the *As* is responsible to generate the genesis block for the BC instantiation.

BlockA*uth* exploits BC as a transaction database shared among all R_i, which are responsible to authenticate $prefix$(es), i.e., IU published by P. The global BC consists of transactions added in sequential order, referring back to the very first one (i.e., when P first registers to the network). The entire BC is private among network nodes, and it cannot openly be reviewed by a non-member. The three essential primitives enabled by blockchain are as follow.

- Retrieve: As per design, a local copy of global BC is available on each R_i. The BC data can be retrieved from any router to perform $prefix$ authentication. Each transaction refers to its previous transaction outputs which are accessed using a previous transaction ID ($previous\_Tx^{ID}$) in the current transaction. Therefore, by retrieving transactions from BC, routers are able to retrieve all the relevant security context used previously for the same $prefix$ authentication [96, 145].
- Adding data is the process which is followed after $prefix$ authentication, however, it resembles to the same process of transferring data.
  - After the $prefix$ is authorized by Ri, a new transaction is accepted in the network. The output of this transaction is referenced by the new input values attached to transaction, named as transaction ID (Tx^{ID}). Along with that the router ID (i.e., Bs ID and CH ID) through which producer is currently attaching and the security context to verify the $prefix$ are added to the transaction. After a successful transaction validation, it is broadcasted in the network.
  - While receiving the transaction, the miner nodes (i.e., LBAs and GBAs) adds the most recent transaction requests into a block. The miners then mines a new block.

  - The miners starts competing to generate the block (later, described in Section 4.5.1.1). When the first miner mines a block, it appends the block to the end of the BC and broadcast the same to other R_i in the network [203, 210].
  - when a new mined block is received, the R_i verify and add it to the local copy of BC.

- Update: The data and transactions are updated at each receiving R_i such that every new block is ordered and linked to the previous block, which makes it impossible for R_i to miss any added information.

4.4.4 Initial Producer Authentication

In BlockA*uth*, mobile producer (P) entails the pk^s_P and sk^s_P key pair, which is associated with the $prefix$(es) owned by P. It is accomplished similar to the traditional mobile networks, i.e., by utilizing mobile device Subscriber Identity Module (SIM) provided to P by the network service providers [16, 68]. The SIM device is hard-coded with the sk^s_P key for P, and it is securely distributed and registered with it's respective pk^s_P key at As. When P connects with the network first time, the As directly verifies the $prefix$ announced by P. To achieve this, P issues the registration interest with the $prefix$ and pk^s_P. The interest also carries a digital signature computed by encrypting $prefix$ and some additional information with P's sk^s_P key. The additional information consists of the ID of Bs and CH with which producer is currently attached, and the IDs related to BC specific fields (e.g., Tx^{ID} and $previous\_Tx^{ID}$). Recall that Tx^{ID} is required to link a Tx to its previous Tx in BC. In particular, Tx^{ID} is the hash pointer of the whole interest message, and it is calculated using a pre-defined hashing algorithm, i.e., $H(.)$, such as Tx^{ID} is the identifier of the current transaction message, where $Tx = H(message)$.

Since the $prefix$ is being requested for verification for the first time, it holds a null value for $previous\_Tx^{ID}$, when an ICN node receives a transaction with $previous\_Tx^{ID}$ set to null, it forwards the transaction (aka IU message) towards As. Once the As receives the transaction, it verifies $prefix$ by comparing and validating the encrypted content using the pk^s_P key of P. After a successful verification of a $prefix$, As generates the genesis transaction (i.e., the first transaction of P in blockchain) for P in the BC and broadcast it in network so that all the miners can update their local copy of the global BC. The genesis transaction includes the following data about P: (i) Tx^{ID}, which also becomes the output of this specific transaction, (ii)

$previous\_Tx^{ID}$, (iii) pk_P^s key assigned to P, (iv) digital signature sent by P in its IU request, and (v) the payload which includes information to increase scalability and efficiency of BC such as Distributed Trust Association metrics (later described in Section 4.5.1.3). Figure 4.3 illustrates the steps taken to perform initial registration of P in our protocol.

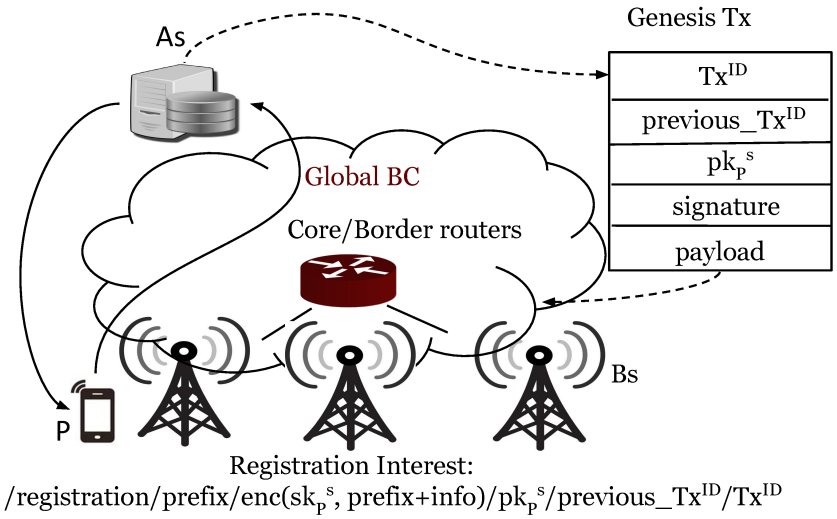

Figure 4.3: Initial authentication

4.4.5 Secure Producer Mobility

In this section, we present the details of IU authentication mechanism at every mobility event, i.e., when a mobile P attaches to a new PoA (i.e., Bs) and issues a new IU following the taxonomy of [32, 225]. Similar to Section 4.4.4, in our IU authentication mechanism, P needs to send IU which comprises of $prefix$ and security content signed with the sk_P^s along with Tx^{ID} and $previous\_Tx^{ID}$.

In BlockA*uth*, we use the PKI-based infrastructure along with Blockchain technology for prefix attestation. The idea of Pretty Good Privacy (PGP) encryption program uses Public Key Infrastructure (PKI) to provide the following three main functionalities (i) confidentiality with encryption, (ii) authentication via digital signatures, and (iii) web of trust via identity validation from peers. We exploit the PKI [83] as a major function to perform $P's\ prefix$ authentication in mobile networks using BC technology. In particular, BlockA*uth* utilizes BC as a database to store pk_P^s, digital signature, and additional information of P to allow each router in the net-

work to authenticate IUs generated by P. In addition, it also validates the integrity of ICN nodes (i.e., R_i) that participate in the BC network.

To securely verify the IU during handover, the producer sign the $prefix$ along with the currently attached Bs and CH IDs with its sk_P^s. Along with it, the pk_P^s, the current Tx^{ID}, and the previously used transaction identifier, i.e., $previous\_Tx^{ID}$, is also sent as plain text. In particular, at each instance when P connects to a new Bs, it issues an IU consisting of the following fields:

- $prefix$[2].
- Public key (pk_P^s) associated with P, which is required in the network for the signature verification of IU.
- Digital signature, which consist of $prefix$ and additional information signed by the private key (sk_P^s) of P, i.e., $enc(sk_P^s, prefix + info)$. The signature ensures $P's$ $prefix$ authenticity and immutability with it's previous authentication event.
- Previous BC transaction identifier ($previous\_Tx^{ID}$), which is used to find previously stored authentication data of any specific P in BC.
- Current BC transaction identifier (Tx^{ID}), it is the hash of the whole IU, i.e., $Tx_{ID} = H(IU)$.

The purpose of signing the Bs and CH IDs along with the $prefix$ is to make the signature immutable with the initial authentic $prefix$ registration, and it also helps to keep track of all previous mobility activities of P which could be used for analysis purposes. In particular, BlockA*uth* aims to achieve the backward secrecy by linking each IU authentication for a P with its previous authentication event in an immutable way through BC, and the process will eventually lead towards the initial $prefix$ registration of P. In particular, on each IU authentication event, a new value of Bs or CH ID[3] is signed along with prefix. However, the $prefix$ for the signature computation remains the same. Therefore, to generate a different signature, the BlockA*uth* utilizes Bs or CH ID as an additional information along with $prefix$. Additionally, the process ensure that current IU message for authentication is received from the same P which has earlier performed a

[2]Note that a plain text name prefix is needed to route the interest towards previous PoA and update the forwarding information.

[3]Since on each handover Bs or CH ID changes, therefore it triggers new additional information.

successful IU authentication. It is because the previous signature composed of previous additional information (i.e., previous Bs/CH IDs) is stored in the BC and immutability is linked with the current transaction through $previous\_Tx^{ID}$.

In our illustration, we only show the authentication steps performed at Bs, however, it has to be considered that each R_i participating in the network (i.e., core and border routers) execute the same authentication steps upon reception of a new IU request. The Bs while receiving the IU, first retrieve and verify the transaction by de-hashing and comparing the attached hash pointer, i.e., $previous\_Tx^{ID}$. The $previous\_Tx^{ID}$ is a hash pointer towards $P's$ previous transaction in BC. In particular, it streamline the task to search the previously chained transaction for that specific IU of P. The router than verifies the $prefix$ by using the pk_P^s, which is stored in header of the previous Tx of P. If the signed content matches the $prefix$ attached in the IU message then the router authenticates the $prefix$ and store the new Tx^{ID} as the hash pointer output for the current transaction.

To illustrate how PKI based IU authentication is performed at each R_i, let $enc(sk_P^s, M)$ be an encryption function which takes input a sk_P^s and an arbitrary string M consists of $prefix$, and additional information, and it generates an encrypted signature M'. The $dec(pk_P^s, M')$ be the respective decryption function which uses pk_P^s. To authenticate the IU, R_i utilize $previous\_Tx^{ID}$ to retrieve pk_P^s from transaction header and then recover M as a function of $M = dec(pk_P^s, M')$. To authenticate IU, R_i matches the content object recovered from M with the original $prefix$ attached in IU. Figure 4.4 illustrates the message flow for initial registration and IU authentication mechanism of BlockAuth protocol in detail.

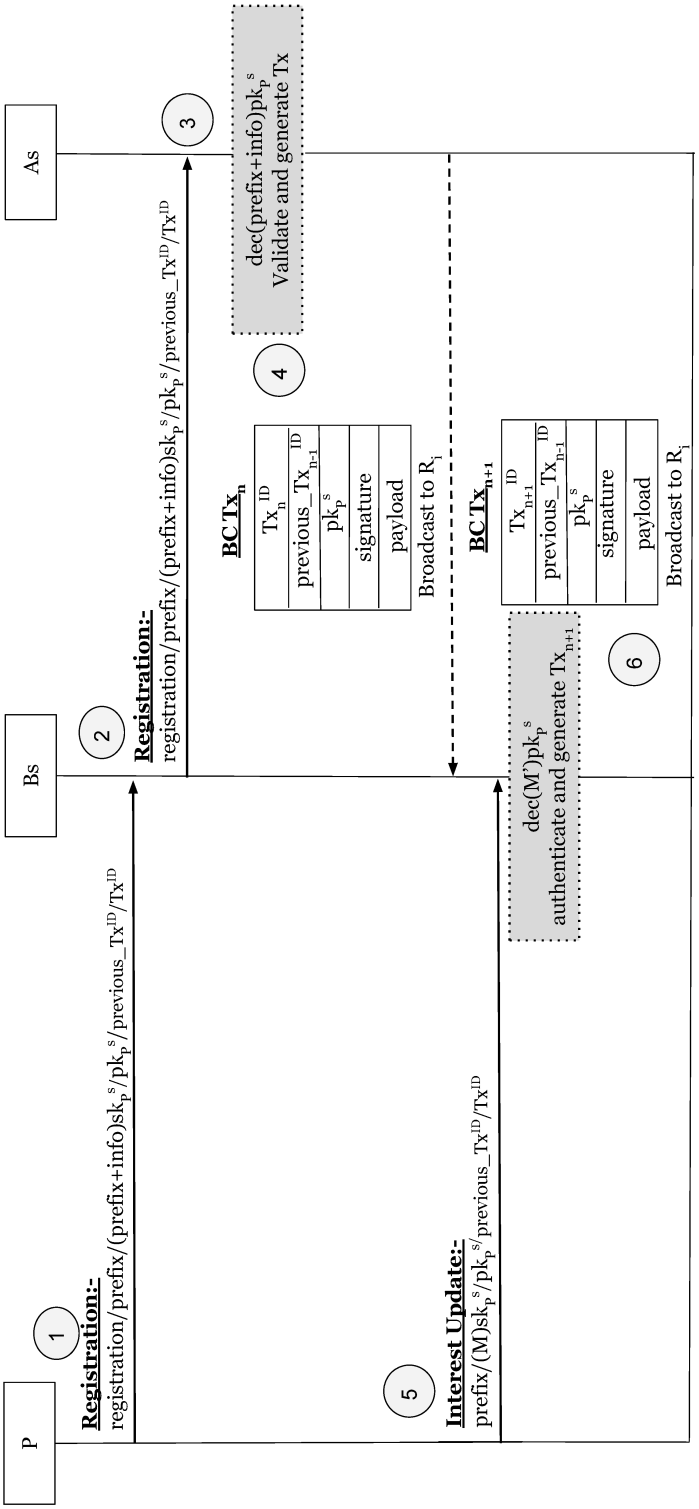

Figure 4.4: Message flow for BlockA*uth*

4.5 Efficient & Scalable BC for BlockA*uth*

In this section, we discuss the mechanism of BC transaction generation and verification in core network and in individual clusters (also referred as global BC and local IL transactions). To exemplify the functioning of BC in two fundamental tiers, i.e., GBC and LIL, we commence our discussion by explaining the following regimes.

- *Transaction:* It is symbolized as a general communication primitive, which is utilized to exchange authentication control information for BlockA*uth*. Recall that as stated prior, the network data flow in our case is separate from transactions.
- *BC Administrator (BA):* It is an entity which is responsible to manage the BC. The key operational task of BA is to verify, generate, broadcast, and store all the received valid transactions. The role of BA functioning in a core network and in clusters consist of slightly different actions which are explained below in detail.

4.5.1 Global BC Transactions

The core network potentially consists of multiple core and edge routers. To ensure scalability, we assume that the subset of core routers and all CHs are managing the global BC, and these routers are named as GBA. In addition, each CH is also functioning as Local IL administrator (LBA). The functioning of LBA is illustrated in detail in Section 4.5.2. The CHs process and manage all transactions that are coming to and from their respective cluster members.

Similar to Bitcoin [151], to ensure integrity of GBAs, the blocks generated by GBAs are secured by means of asymmetric encryption, digital signatures, and cryptographic hash functions (e.g., SHA256). In contrast, for a transaction to be considered valid, the protocol requires only a single signature transaction, which is the signature of the requester, i.e., P. The structure of a transaction in BlockA*uth* is shown in Figure 4.5. The foremost field in the structure is an identifier for the current transaction, which is the hash of the entire IU message[4]. The second field is a hash pointer towards the previous transaction of the same P. In this way, all the transactions generated by each P are chained together, and it is followed by the public key and signature of the mobile P, i.e., $prefix$ and respective information

[4]It may optionally exclude to take the hash of $prefix$ which is always consistent and attach without encryption for routing purpose.

signed by the private key. The additional data is stored in the fifth field of the payload of Tx. This additional data helps to create the reputation of the routers which are performing as BAs. It contains the output of the transaction which is set by the receiving BA. The output fields consist of the following two entries: (i) the total number of transactions generated by a sending BA, which are been accepted by the receiving BA, and (ii) the total number of transactions rejected by the receiving BA. In particular, the data in both the fields provide previous information that is utilized to compute the reputation of the BA processing the transactions in the network. A *Distributed Trust Algorithm* use this specific data to optimize the efficiency of BC [71] while updating related communication messages. The algorithm for the same is outlined below in Section 4.5.1.3.

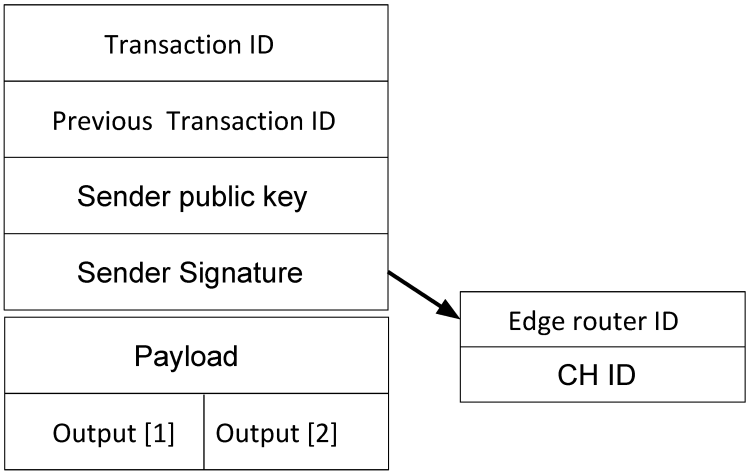

Figure 4.5: Transaction structure

The transactions in the core network follows the genesis transaction which is generated during the initial registration of P by the As. Therefore, each GBA and CH must first append a genesis transaction received from As that attends to be a starting point for this particular P in global BC.

Each block in the BC comprises of two core portions namely, transactions and block header. The header of the block contains the hash of previous block, block generator (i.e., BA) ID, and signatures of the block generator. The hash of the preceding block in the BC safeguards immutability. If an adversary attempts to corrupt any previously stored transaction in BC, then the hash of the subsequent blocks that are stored on top of it will no longer be consistent with the global BC. Hence, it will expose the attack. Similar to Bitcoin, multiple transactions are grouped together and then processed

as one block. A block can store at most Tx^{Max} transactions. The value of Tx^{Max} affects the BC throughput.

When CH receive a transaction (say x), it first checks whether the P that generates the transaction has moved within the same cluster or not. It is achieved by forcing the P to issue two distinct pair of transaction IDs, one related to local IL (Tx_i^{ID}) and other to global BC (Tx_j^{ID}). The CH (also functioning as GBA) will first perform a look-up for Tx_i^{ID} in local IL. In case, the local IL transaction ID has no pointer found for previous transaction, the CH uses the global transaction ID to retrieve the previous immutable transaction generated by the P in global BC. In contrast, if the incoming transaction matches the hash pointer for Tx_i^{ID} in local IL, then CH processes the transaction and updates the local IL among the cluster members. The procedure of managing the local IL is later detailed in section 4.5.2.

If a transaction (say x) does not belong to the BA cluster, then the transaction is verified using Tx_j^{ID} and broadcasted to all other GBAs. All global transactions are being verified by each router and are stored in a local unprocessed transaction pool at each GBA. When the size of the transaction pool reaches to Tx^{max} then the GBA creates a block using the transactions in the pool, and it starts the validation process which is followed by the block inclusion in the global BC using a consensus algorithm.

4.5.1.1 Consensus algorithm

Instead of using traditional resource-intensive consensus algorithms such as PoW or PoS, BlockA*uth* exploits a time-based consensus algorithm given in [70, 71]. The consensus algorithm ensures that a block generator is nominated randomly among all miner nodes (i.e., BC administrators) that are participating in the protocol. Moreover, the block generator is limited to the number of blocks a node can generate within a particular duration. In order to enforce randomness among block generation to avoid forging, before generating a new block every single BA administrator is forced to wait for a random time known as *waiting-period*. Due to different waiting-period experienced by each BC administrator, an administrator might receive a new block created by another administrator which contains some or all of the transactions currently present in its own pool of transactions. Therefore, in this case, the BC administrator receiving the new block will remove all those transactions from the pool that has been already stored in the BC by the recently added block(s).

By forcing the block generators to wait for a random time reduces the duplication of blocks, which can be generated simultaneously. The maximum

waiting-time is capped at twice the maximum end-to-end delay between the routers in the core network. The cap is used to ensure that there is sufficient time for disseminating a newly generated block by other BC administrators. When a new block is generated, it is broadcasted to all other routers so that it can be appended in their local copy of global BC.

To protect the overlay against a malicious BC administrator which can potentially generate a large number of blocks with fake transactions that leads to an appending attack (later discussed in Section 4.6), the periodicity with which an administrator can generate new blocks is restricted such that only one block can be generated over an interval denoted by the consensus-period. Any non-compliant blocks are discarded and the trust associated with the responsible BC administrator is decreased as outlined in the Section 4.5.1.3. The consensus-period is adjusted by *Distributed Throughput Management* (DTM) which we discuss in Section 4.5.1.4.

4.5.1.2 Verification

Every BC administrator validates each receiving block before appending it to its local BC. To validate the block, the signature of the block generator is validated. It is assumed that each BC administrator uses a pre-defined asymmetric keys[5] for block generation and communication. Each individual transaction in the block is also verified, therefore, similar to Bitcoin, a block is considered to be valid only if all transactions in the block are valid.

Algorithm 7 outline the procedure for verifying an individual transaction (say x). As discussed in Section 4.4.5, the link between the successive transactions of a P is established by including the hash of the previous transaction, which was generated by P in its previous transaction as Tx ID. Thus, the BA first confirms this linkage between consecutive transactions of a P by comparing the hash of its previous Tx ID in x with the first field of previous transaction (i.e., $x-1$). Following this, the signature of the P that is contained in the fourth field of x is verified using its pk_P^s in $x-1$. Further, the signed payload containing the sender prefix is compared with the prefix attached with the interest. If the steps are completed successfully, x is verified.

Recall from Section 4.5.1 that output [0] and output [1] in the payload are used to create the reputation of BA that is sending the x. Initially, the sending BA set these outputs based on its history of transactions. If the receiving BA (i.e., GBA and CH) accept a transaction, then it will

[5]We propose a solution implementing private BC, therefore BC designer, i.e., network owner, is responsible for key distribution in core network.

Algorithm 7 BlockAuth Tx verification

```
1: procedure TX_VERIF_PROC(prefix, pk_P^s, M', Tx^ID, previous_Tx^ID)
2:     Output: True or False
3:     if previous_Tx^ID ≠ Tx − 1^ID then
4:         return False
5:     else
6:         if enc(sk_P^x, M) ≠ dec(pk_P^{x−1}, M') then
7:             return False
8:         else
9:             check prefix with M'
10:            return True
11:        end if
12:    end if
13: end procedure
14: close;
```

increase the output [0] by one. Otherwise, it increases the output [1]. To protect the BC against routers that claim false reputation by increasing their outputs before sending them to the BA, in the next step of transaction verification, the receiving BA checks that only one of $x's$ outputs, i.e., either the number of successful transactions (i.e., output [0]) or the number of rejected transactions (i.e., output [1]) is increased by one.

4.5.1.3 Trust association among BC administrators

For the network routers participating in BC formation, verifying all transactions and blocks is computationally challenging. Particularly, when the number of macro-mobility events (i.e., inter-cluster) in the network increases. In our case, routers do not need to verify complete BC instantiation at each instance. Still to preserve the required immutability and to ensure scalability considering smooth handover requirements, we exploit *Distributed Trust Algorithm* (DTA), which gradually reduces the number of transactions needed to be verified by BC administrator in each new block. It is achieved by building a trust relation between the routers generating the new blocks, i.e., BC administrators. The trust algorithm builds the concept of primary and secondary evidence between BC administrators as follows:

- Primary evidence: The evidence in which a BC administrator (say A) has at least one previously verified block generated by another administrator (say B) is called as a primary evidence of B for A.

- Secondary evidence: If BC administrator A does not have any primary evidence about B, however, anyone among other BC administrators has established that the block generated by B is valid, than A has a secondary evidence about B.

All BC administrators maintain a list, which stores the relevant information regarding primary and secondary evidence. For instance, the administrator record the list of blocks and senders that are verified. Therefore, considering the attacks in which any administrator might generate the blocks that are not compliant with the proposed consensus algorithm results in decreased trust association. Hence, for one discarded altered block, the receiver decrease the trust association by a factor of one for the malicious sender. If the malicious BC administrator remains with this conduct, it faces with consistently reduced trust rating, which results in more and more of its transactions being verified before accepted for further processing by the BCAs. Vice versa, for one accepted verified block, the receiving BCA increase the trust association of sender by one. In the case of secondary evidence, the administrators check the number of other administrators that have verified the received block to get the trust association of block generator. The key benefit of the trust algorithm is that stronger the evidence is received for an administrator generating blocks, lower the number of transactions is the blocks generated by it that needs to be verified before adding into the BC.

Aiming at the performance and objectives of BlockA*uth*, Figure 4.6 elaborates the functionality of trust association algorithm, which a BCA follows while verifying the transactions of a block received by another BCA. In particular, the primary evidence takes precedence over the secondary evidence. For instance, if a BCA has the primary evidence about any block generator, e.g., 50 earlier received blocks by a BCA are verified, then only fraction of transactions within the current block are selected for verification[6] (refer to Figure 4.6). Correspondingly, in case where no primary evidence is available for the block generator, the BCA first check for the availability of secondary evidence. The secondary evidence indicates the percentage of other BCAs that have vouched for the block generator in question, e.g., in case only 20% of other BCAs have assert the block generator then 80% of the transactions within the received block needs to be verified (refer to Figure 4.6). Finally, if no evidence is recorded, then all the transactions in the block are verified. Note that certain fraction of transactions are always required to be vali-

[6]The selection of transactions within a block can be random to make trust association more robust.

dated even if there is a strong evidence, it is to protect against any potential malicious BCA.

Primary evidence	Previously validated blocks	10	20	30	40	50
	Transactions Required to validate	80 %	60 %	40 %	20 %	10 %
Secondary evidence	% of BAs signed the block	20 %	40 %	60 %	80 %	100 %
	Required to validate	80 %	60 %	50 %	40 %	30 %

Figure 4.6: Distributed Trust Association among BA

4.5.1.4 Distributed throughput management

To strengthen the throughput performance of the proposed BC solution, we make use of a *Distributed Throughput Management* (DTM) mechanism [70, 71]. The BC throughput is measured with the number of transactions added in BC per second. In our proposed protocol, we make sure that BC throughput should maintain a desirable range since each macro mobility event result in a new global transaction. The DTM monitors the BC utilization at the end of every consensus-period by computing the ratio of total number of new transactions generated in the network by mobile producers to the total number of transaction added in the BC. Note that, since all transactions and blocks are broadcast to all BC administrators, the utilization computed by all administrators is similar. The BC throughput can be accustomed by two ways: (i) changing the consensus-period time, it indicates the frequency with which blocks are appended to the BC, and (ii) changing the number of BC administrators generating the blocks as each administrator generate single block within a consensus-period.

To better illustrate DTM, let's assume that μ is the BC utilization factor that BCAs calculate at the end of each consensus period, and the aim of DTM is to ensure that μ remains in the certain acceptable range to meet

the requirements set by network operator (i.e., $\mu_{min} > \mu > \mu_{max}$). From [71], we calculate μ as follow.

$$\mu = \frac{N * R * consensus - period}{Tx^{max} * M}, \tag{4.1}$$

where N and M denotes the total number of network nodes and the number of nodes assigned with the functionality of BC administrator by network operator. R denote the average rate at which BC administrators generate new transactions per second. In particular, R is estimated through total number of transactions generated with in a consensus period. Equation 4.1 illustrate two ways through which μ can be adjusted to put it in the desired range of μ_{min} and μ_{max}, i.e., changing consensus time or M. For instance, if μ exceeds the desired maximum range (i.e.,μ_{max}), then DTM first checks that if consensus-period can be reduced. For that DTM compute a new value of consensus-period using Equation 4.1, which take the μ equal to the mid-point of desired range, i.e., between μ_{min} and μ_{max}. It results in a stable operating point in the performance of the network's transaction throughput. Conversely, if the consensus-period can not be reduced then the network should be optimized to increase the number of BCAs. The new value for M is also calculated similarly using Equation 4.1 and by taking the value of μ as a mid-point of the desired range. In addition, the consensus-period is used with the default maximum value while calculating M. This feature allows BlockAuth to scale in an efficient manner for later optimizations. With increase in number of M, the transaction throughput would increase and the max value of consensus-period can be further utilized for throughput enhancement. In case, when the utilization factor drops to the minimum value, i.e., μ_{min}, the similar, but inverse approach is adopted by DTM. First, DTM attempts to increase the consensus-period to optimize μ, else, it should decrease the number of BCAs, i.e, M.

4.5.2 Local IL Transactions

Each cluster is comprised of various mobile producers connected to it. The individual cluster is managed by a Local BC Administrator (LBA). The local transactions are encrypted with asymmetric encryption which is done using a lightweight cryptographic hash function. The process uses a PKI infrastructure similar to global BC. Recall that PKI structure is predefined by network owner using *As* server and SIM, which stores private keys. In each cluster, the LBA centrally manages the local Immutable Ledger (IL) whose structure is similar to BC, and it processes local transactions that are generated and propagated within a cluster. In addition, the global trans-

actions that are generated to or from the cluster are also managed by the same CH when a producer initially enters into a cluster.

The local IL records all local and global transactions of the P for which the LBA is the only BA. In particular, if P remains in one cluster, CH will maintain both local IL and global BC transactions of the P. As described earlier, each block in the local IL also contains a block header and a policy header. The block header maintains the hash of the previous block to ensure immutability similar to the global BC as discussed in Section 4.5.1. The policy header is in the form of an Access Control List (ACL), which define rules for processing the local and global transactions. Each producer while roaming in the same cluster uses local IL transaction ID pair, i.e., Tx_i and $previous\_Tx_i$. In particular, the local IL works similar to the global BC, but once being authenticated in the cluster, the producer use and update only local transaction IDs until it moves to another cluster. The first instant of the local transaction ID refers to the genesis output generated by the As. This is performed by the respective CH (i.e., LBA) when a producer initially joins the cluster. Therefore, the structure of the local transactions in local IL is also similar to global transaction. P uses the local IL to process seamless handover while roaming within the cluster to reduce communication overhead in the core network. The procedure followed by the P in macro and micro mobility scenarios during prefix authentication mechanism is discussed below in detail.

4.5.2.1 Intra cluster handoff

In this type of handoff, BlockA*uth* exploits the location change of P within the same cluster which is controlled only by CH (i.e., also denoted as soft handoff). As mentioned earlier, each producer and CH extracts its own value of Tx^{ID} when referring to the switch within the same cluster, i.e., Tx_i^{ID}. The mobile P can identify its attachment to recent cluster by examining the CH ID. This is achieved during the scan process of handoff. In particular, the handoff procedure starts with a neighbour discovery phase called *Scan*, through which mobile host acquires the cluster ID [163, 192]. In this way, after being authenticated in the same cluster, the producer only update and send the local Tx_i^{ID} chain for being authenticated again in the same cluster. The local IL is administrated and mined by the respective CH. The CH checks the local Tx_i^{ID} to examine, if the P has not changed the cluster and finds the immutability in the local IL. The transaction is then processed similarly as described previously. After verification, the local blocks are processed and broadcast to the rest of the Bs within the same cluster.

4.5.2.2 Inter cluster handoff

For inter cluster mobility event (also named as hard handoff), P update and utilize the global Tx_j ID pairs, which are being used in the previous cluster. The Bs in the new cluster could not find the immutability in the local IL, and therefore, verify the transaction using global Tx_j ID. The transaction is than processed by the respective CH, and later it is stored into global BC.

Table 4.2: BlockA*uth* security analysis against various threats

Threat	Description	Mitigation
Prefix hijacking attacks	Divert traffic of legitimate users and network towards hijacked addresses (prefixes)	Each router in the network authenticates the prefix (IU) to verify the ownership before updating and forwarding the state of network (see section 4.4.4 and 4.4.5).
Appending attack	Adversary can compromise a BC administrator to generate false blocks and transactions, it leads to corruption of forwarding information in the network	BC administrator is able to detect a fake block during the verification step (see Section 4.5.1.2), and therefore, it can identify the malicious BCA.
Denial of Service attack (DoS)	Adversary floods router with fake transactions to overwhelm the node such that it cannot devote any resources to process genuine transactions	Unlike [51], in our proposal the router executes IU authentication mechanism once for a verification process and it does not entails the invalid IU in its forwarding state (see Section 4.4.5).
Distributed Denial of Service attacks (DDoS)	Adversary attacks on multiple edge router and BCA to flood the network with fake transactions generated from multiple sources	The use of asymmetric keys between BAs make it impossible for an adversary to initiate the DDoS attack. In addition, each transaction and block in the network is verified by the global consistent and immutable image of BC (see Section 4.5.1).
Replay attacks	A legitimate producer issues the same IU for prefix authentication from two different access points to corrupt the network forwarding information	BC mitigates the replay attack as a single transaction output is immutable to hash pointers. The router selects only the latest valid IU message from the producer.
Packet discarding attack	The BC administrator or CH discard transactions which are being received to and from the cluster members	A cluster member can change the CH or BC administrator it is associated with, if it observes that its transactions are not being processed.
False reputation	Any malicious BC administrator tries to increases its reputation	Other BC administrators can detect false increase during transaction verification.

4.6 Security Analysis

In this section, we present the evaluation of our proposed protocol with qualitative security analysis by considering the adversary model elaborated in Section 4.4.2.2. We show, in various scenarios, that an adversary is not able to successfully initiate the IU mechanism for the $prefix$(es) which she does not own. We assume that the adversary can be any node functioning in the network including BC administrators, CHs, and mobile producers. A legitimate P can also act as an adversary (Adv) to re-use the assigned security credentials to perform prefix hijacking or sybil attacks. Moreover, we assume that Adv is capable to sniff communications, generate false transactions and blocks, discard legitimate transactions, analyze multiple transactions in an attempt to deanonymize a node, and sign fake transactions to legitimize colluding nodes. In our model, we use standard secure asymmetric encryption, digital signatures, and cryptographic hash functions (e.g., SHA256), which cannot be compromised by the Adv.

4.6.1 Mitigating Prefix Hijacking Attack

The initial registration phase can be passed by an adversary if it can insert false transaction in the BC. For this purpose, an Adv need to have a valid signature along with the $prefix$ for the registration of the interest. The computation of signature is with unique private key of the P, which is registered with $prefix$ owned by it. Since the private key of P is never transmitted over the network and it is always stored on the device(s) given by the network owners to the producers, e.g., SIM card. Therefore, Adv is not able to generate a valid registration interest without knowing the private key of the P. The only case, to pass the initial registration process is to replay a valid initial registration interest. In this regard, if replayed interest has received after the As has already received the valid interest, then the malicious interest is discarded by the As, resulting in failed authentication. It is because the hash pointer for current transaction output requires a new transaction output, which is immutable to previous transaction as elaborated in Section 4.4.5. The only case in which the replayed interest can pass initial authentication is, if the valid interest is being received after the replayed one such as due to network congestion or by exploiting signal jammer. At this stage, Adv must generate a valid IU subsequently to update the forwarding state of the edge routers at each mobility event.

To express a valid IU for $prefix$ of P, the adversary must be able to generate a new valid signature including prefix and PoA information and

previously used immutable hash pointer towards the previous transaction, i.e., previous Tx^{ID}. Along with $prefix$ name, the complete security content attached (refer to Section 4.4.5) are computed using signature-based encryption and cryptographic hash function (e.g., SHA256). The security properties of PKI-based blockChain methodology makes it impossible for an adversary to generate a valid IU, which is immutable to BC ledger without knowing the private key, cryptographic hash function, previously used immutable transactions pointers, and previous PoA's information.

In Table 4.2, we summarize specific security attacks to which mobile networks and BC are particularly vulnerable and outline how our proposed protocol defends against them. We also analyze the robustness of BlockA*uth* against each of these attacks and the possibility of the attack to happen based on European Telecommunications Standards Institute (ETSI) [14] risk analysis criteria.

4.7 Performance Evaluation

In this section, we evaluate the overhead introduced by BlockA*uth* on ICN routers. Our evaluation mainly focus on the IU authentication mechanism since it is needed at every mobility event, and it is the most demanding step of our protocol. In particular, we provide an analysis of BlockA*uth* regarding: (i) computational overhead, and (ii) additional storage cost introduced at routers.

We compare BlockA*uth* with the hash-based verification approach that is adopted in most of the prefix attestation proposals [53]. We consider the processing steps to be same in both approaches (i.e., both issue an Interest Update that will be verified at each router). In the hash-based verification, an IU carries a hash value, while in BlockA*uth*, an IU carries the security content to verify and initiate a BC transaction.

4.7.1 Computational Overhead

To evaluate the computational overhead introduced by BlockA*uth* on the routers, we compute the time required to perform the IU authentication with both hash-based and signature-based approaches. Then, based on the analytical model proposed in [90], we compute the impact of BlockA*uth* on overall router's throughput with increase in the producer mobility rate. Note that from the term router throughput, we mean the number of regular interest packets processed by a router excluding the Interest Updates. In particular, we compute the delay occurred for the authentication of mobil-

ity messages, i.e., IUs, then considering the delay we compute the router's original throughput with increasing producer mobility.

The time required to authenticate the IU is defined as the sum of the time required to retrieve the relevant transaction from BC (i.e., security context in case of hash-based approaches) and the time required to authenticate the $prefix$. Considering the fact that the BlockA*uth* uses a private BC, the time required to retrieve the relevant security content, i.e., previous transaction using hash IDs, is negligible[7].

To compare the performance of BlockA*uth*, we evaluate BlockA*uth* and hash-chain based protocols [53] by considering the hardware installed on the Bs. We conduct the evaluation on EPYC 7601-AMD processor as a reference hardware, and we get both hash-based and signature verification time from [36] as a benchmark. Table 4.3 and Table 4.4 reports the time required to verify message (i.e., 59 bytes in size) using the public-key signature-based and the hash-based schemes, respectively. The security context is stored together with the forwarding state in the corresponding table, and it can be retrieved during the regular lookup using transaction ID[8]. Therefore, in BlockA*uth*, the time delay to authenticate the IU is the signature-based verification time and it is the only dominating factor. Table 4.3 and Table 4.4 reports the time required for the verification using various public-key signature and hash-based cryptographic schemes, respectively. In particular, it reports the results collected in ECRYPT Benchmarking of Asymmetric Systems (eBATS) for public-key signature systems [36]. To compute authentication delay for each cryptographic scheme, we used median values of cycles that are required to verify 59 bytes with specified processor. We then analyze the impact of the BlockA*uth* authentication delay on edge router throughput. To this end, we compute the routers throughput with increasing producer mobility rate.

Table 4.3: Authentication delay PK signature based schemes

Public key signature based	**Delay**	
Name	**1024-bit**	**2048-bit**
RSA	0.20 μs	0.34 μs
DSA	2.01 μs	6.25 μs

To compute the route'r throughput, we utilize the model illustrated in [53, 90]. Let's assume that σ is the ratio of IUs over the total num-

[7]For hash-based approach security context is the recent hash value with corresponding sequence number [53] [53, 90].

[8]The previous transaction ID of the producer is the relevant security context

Table 4.4: Authentication delay hash based schemes

Hash chain based	**Delay**
SHA256	0.0019 μs

ber of normal packets received at a router's ingress interface. Thus, the router throughput can be defined as a ratio of packets/second (λ), which can be calculated as follow.

$$\lambda = \frac{1-\sigma}{\tau_{process} + (\sigma * \tau_{authentication})}. \tag{4.2}$$

In Equation 4.2, $\tau_{authentication}$ is the average authentication delay for verifying the prefix, and $\tau_{process}$ is the average processing time that a router takes for an ordinary packet processing. We consider that maximum throughput of edge router, i.e., Bs, to be 0.50Mbps. Therefore, $\tau_{process}$ is calculated to be 2μs. To compute the impact of BlockA*uth* on router throughput, we apply the values reported in Table 4.3 and Table 4.4 in Equation 4.2.

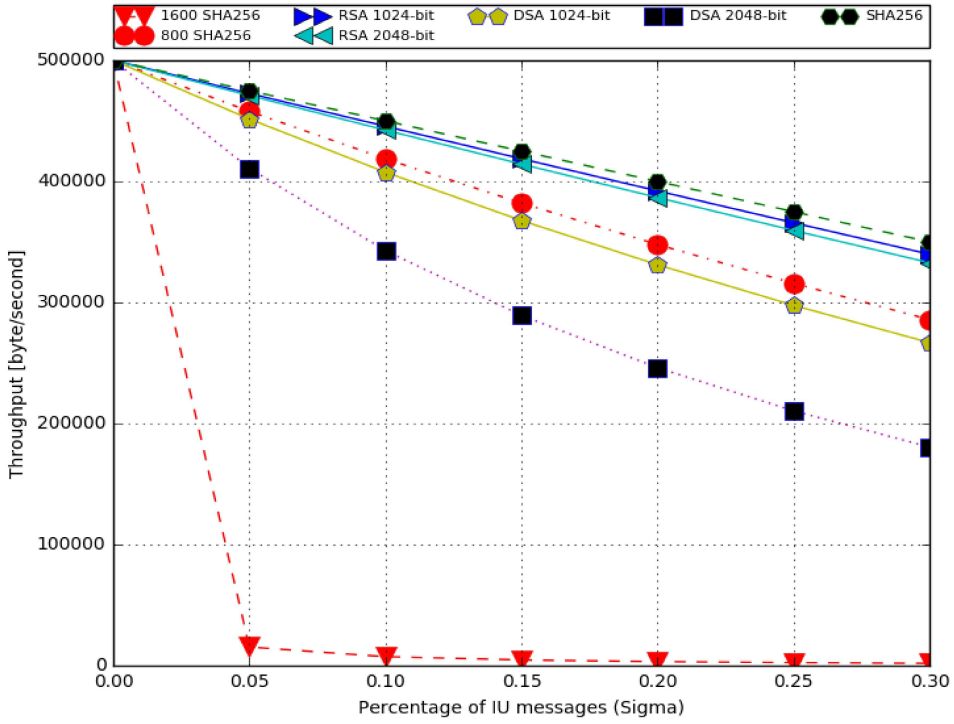

Figure 4.7: Edge router throughput

Figure 4.7 depicts that BlockA*uth* exhibit comparable performance to the hash-chain based verification, in case when there is no significant DoS attack. The result shows that the BlockA*uth* is able to provide approximately 90%–94% of the router's original throughput (i.e., without IUs authentication), when mobility rate is upto 5%. In precise, router is able to maintain almost 94% of the original throughput with encryption schemes, e.g., RSA 1024-bit, when 5% of the received traffic includes IUs, i.e., traffic triggered by mobility events. Similarly, for 10%-15% increase in mobility traffic (i.e., IUs), BlockA*uth* is able to maintain 76% to 90% throughput, the percentage value depends on the cryptographic scheme applied (as shown in Figure 4.7). For instance, while using RSA, BlockA*uth* is able to maintain edge router throughput (i.e., 88%) and it can perform IU authentication on line-rate when mobility rate is upto 20% (refer to Figure 4.7). Moreover, even with the most optimal mobility scenario where a router receives 30% of IUs in overall traffic, the maximum throughput achieved by router is 75% (i.e., with RSA 1024-bit). In summary, results report that network designer can choose the most appropriate cryptographic scheme depending on the network conditions which relates to the frequency of mobility events.

Figure 4.7 also shows the robustness of BlockA*uth* against the DoS attack which is provoked by the one-way hash-chain based prefix authentication proposals [53] (refer to Section 4.2). We calculated the router throughput for hash-chain based protocol under DoS attack where a router is force to compute hash (i.e., SHA256) for 800 and 1600 times to detect a legitimate IU and in the process the throughput decreases significantly as it is shown in Figure 4.7. The figure depicts noticeable degradation in the throughput where the hash-chain based mechanism (i.e., SHA256) needs to compute just 1600 hashes per IU for authentication. In particular, just with 5% increase in the mobility, the router's throughput decreases nearly to zero during DoS attack. On the other hand, BlockA*uth* efficiently mitigate the DoS attack as it does not require to reach to any synchronized state of the hash chain values.

4.7.2 Additional storage cost

The storage cost introduced by the BlockA*uth* relates to the size of the blockchain. Each router participating in the protocol stores the blockchain along with the forwarding states. After the initial registration of the mobile producer, the size of the BC grows with each new transaction that has been verified and added in it. This relates to the instance when producer issues a legitimate IU at each mobility event. Thus, the additional storage cost

introduced by the protocol can be computed as follow.

$$storage\_cost = N_\sigma * size\_Tx. \tag{4.3}$$

Here N_σ is the number of mobility events initiating IU messages, and $size\_Tx$ is the size of a single transaction needed to perform $prefix$ authentication. For BlockA*uth*, we assume that the size of the Tx is 59 bytes [36] when using both RSA and DSA. For hash chain based authentication, we assume the size of security context to be 32 bytes [53].

Figure 4.8 shows an increase in storage cost with respect to the number of mobility events in the network. In general, a mobile EPC network consists of mobile users in the order of 1 million. If we consider a scenario where each mobile producer is initiating a IU message, i.e., every producer has triggered a mobility event, then we can observe that the storage cost is about 60MB for each router. Current routers can easily store such amount of data due to availability of storage memory. In addition, network owner can optimize BlockA*uth* by using an efficient data pruning technique.

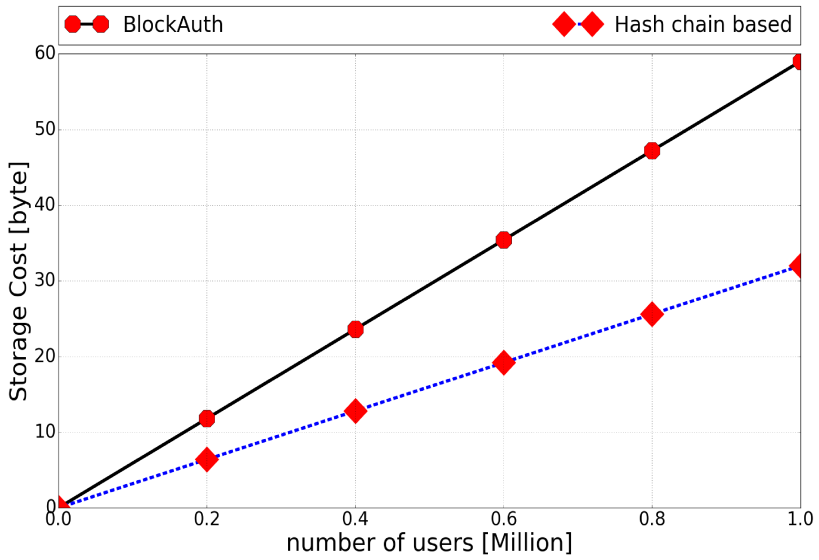

Figure 4.8: Additional storage cost at each router

4.8 Summary

In the context of ICN, the practice of producer/consumer communication model primarily appreciates seamless mobility support to mobile nodes. It is due to the result of decoupling time and space among request resolution and

content transfer. However, the dynamic interaction between producer and forwarding plane in ICN introduces new security challenges in the network. For instance, a producer could send a false interest update request in the network and all the ICN routers that receive such a request will update their forwarding table accordingly, thus leave the forwarding table in inconsistent state.

In this chapter, we investigated and proposed solution to the security challenges related to producer mobility tracing-based protocols. Particularly, to mitigate prefix hijacking attacks and resolve security and privacy issues in ICN mobility management. We presented an efficient blockchain based distributed prefix authentication protocol, which offers reliable and faster mobile user authentication. We show that our proposed protocol is completely distributed, lightweight, and it can be easily deployed on different network access platforms (e.g., 4G, 5G, and WiFi). The security and performance analysis shows that our proposed protocol performs significantly better when compared to state-of-the-art, and it efficiently mitigates prefix hijacking, Denial of Service, and other telecommunication networking related attacks. In addition, our approach is able to maintain the router's original throughput upto 94% (i.e., able to perform prefix authentication at line rate). In terms of storage, it is able to handle billion of mobile producers just by consuming tens of megabyte on each router.

Chapter 5

Denial of Service Mitigation in ICN

One of the fundamental goal of ICN architecture is "security by design". In contrast to existing TCP/IP based Internet, where security issues were (and are still being) identified along the way, the ICN design focuses on both awareness of issues and support for features and countermeasures from the outset. Nevertheless, attackers have exploited the ICN's architectural features to build novel attacks and introduced a new type of Distributed Denial of Service (DDoS) attack, better known as Interest Flooding Attack (IFA).

Regardless of the substantial quantity of research on ICN security, we identified that the proposed defence mechanisms for IFA [19, 49, 60, 87, 172] have one or more of the following limitations. First, the legitimate traffic is likely to be damaged, since most of the proposed countermeasures [19,49,201] limits the rate of incoming traffic and are not able to differentiate between legitimate and malicious packets, thus resulting in unfair punishments. Second, since each router has to perform first an attack detection and then attack mitigation, during the first phase (i.e., inaccurate), most of the approaches are likely to encounter harmful consequences. Finally, the proposed collaborative mechanisms [19, 49, 60, 172] introduces unnecessary overhead given by the extra messages exchanged among routers.

In this chapter, we focus on efficient IFA mitigation in ICN. We propose an efficient mechanism, named as *Choose To Kill IFA (ChoKIFA)*, which mitigates the damages caused by IFA by differentiating the malicious traffic from the legitimate one, and by reducing the former, without any collaborative communication or global network monitoring. In order to do

so, ChoKIFA exploits the Active Queue Management (AQM) scheme, i.e., CHOose and Keep for responsive flows, CHOose and Kill for unresponsive flows (CHOKe) [160], and without any delay penalizes the malicious traffic by dropping both the new incoming malicious interests and removing the ones already stored in the PIT. Thus, routers are able to independently detect and mitigate the attack in progress as-soon and as-close to the adversary as possible, while maintaining the simplicity of forwarding. We evaluate the effectiveness of ChoKIFA through extensive simulations on ndnSIM simulator [20], and by comparing it with the state-of-the-art mitigation approaches [19]. The results show that ChoKIFA effectively mitigates the adverse effects of IFA in the network. In particular, ChoKIFA is able to guarantee legitimate interest satisfaction rate up to 97% and it shows up to 40% less false positives in comparison with rate limiting mitigation approaches.

5.1 Interest Flooding Attacks (IFA)

Similar to IP packets, interest packets in ICN also consumes sensible share in the network capacity. Thus, a massive amount of interest packets might lead to congestion in the network and results in legitimate traffic to be dropped. If a well-coordinated DDoS attack targets a specific name prefix, it can concentrate malicious traffic in the certain segments of the network since the routing in ICN is grounded on the name prefix. As stated in Section 1.2, routers in ICN maintains per-packet state for each interest packet in PIT. Therefore, the immense amount of malicious interests can result in exhaustion of routers memory and resources, and prevent them from creating PIT entries for new incoming traffic, resulting in the disrupt of benign users services. In general, IFAs are categorized on three types based on the type of content requested by adversary [87]:

- Type I: existing or static content.
- Type II: dynamically generated content.
- Type III: non-existent content.

In Type I, several zombies from multiple locations generate large number of interests for an existing content which propagates through all intervening routers caches. In result, interests (i.e., legitimate) for the same content are to able to reach the producer(s) since they are being satisfied by the cached

copies. In particular, this type of attack is quiet restricted since in-network content caching provides a built-in countermeasure. In Type II, adversary issues dynamic requests for existing content, therefore, all interests packets are propagated towards the producer(s), resulting in bandwidth consumption and PIT exhaustion. Correspondingly, targeted producer waste considerable computational resources due to signing the content (i.e., per-packet operation) which is itself expensive. In Type III, adversary requests unique non existent (unsatisfiable) content. These interests, that cannot be collapsed by routers, are routed towards the producer(s). Such interest packets consume memory in routers PIT until they expire due to interest life-time. Therefore, a massive number of non-existent interest packets in the PIT table leads to benign interest packets being dropped in the network.

In this chapter, we focus on the IFA where adversary generates unsatisfiable interests. We consider routers and legitimate traffic to be primary strategic victims of this attack. Using a valid name $prefix$, there are many ways for an adversary to generate these unsatisfiable interests, e.g., (i) by enabling the name of the interest to: $/prefix/nonce$, where the suffix nonce is a random value. Such interests are propagated throughout towards the producer and are never satisfied. (ii) By swapping the `Publisher Public Key Digest` field to a random value. Subsequently, no public key would match this value, therefore, will never be satisfied. (iii) Lastly, by setting the `Interest Exclude filter` to exclude all existing content starting with $/prefix$. In consequence, the interest can never be satisfied as it concurrently requests and excludes the same content. In the rest of this chapter, we use the over-all term IFA refer to as the above described type III of IFA.

5.2 Related Works

In this section, we describe the existing solutions for IFA mitigation in ICN. Then, we illustrate the role of various active queue management schemes mitigating DDoS attacks in existing IP architecture.

5.2.1 Solutions mitigating IFA in ICN

Afanasayev et al. [19] proposed four different methods to deal with IFA. The first method introduces a simple limit on the interfaces based on the physical capacity of the links, resulting in underutilization of the network [19]. The second method is a slight alteration of token bucket algorithm providing per-interface fairness. The algorithm regulates the number of outgoing interests by limiting the assigned tokens to a specific outgoing interface. The

major drawback of this method is that it does not discriminates between benign and malicious interests while assigning the tokens, and relatively admits the number of malicious interests. Therefore, the benign ones are also dropped due to the fair distribution of tokens. The third method is based on the per-inteface ratio between interest and their corresponding data packets, namely satisfaction-based Interest acceptance. In this method, tokens are distributed fairly between each incoming interface according to their interest satisfaction ratio. The drawback of this method is that each router decides to forward/discard interest(s) using its local estimation of interest satisfaction ratio. Therefore, the probability of benign interests being forwarded declines as the number of hops between the consumer and the producer increases [19]. The last method is a collaborative approach called as satisfaction-based pushback. In this case, each router sets an explicit limit value for each incoming interface, and announce this value to all downstream routers. This method has shown to be more effective than the previous two, but the legitimate stream is still influenced, especially when the path is long. Moreover, it creates unnecessary signalling overhead in the network.

Furthermore, several defense mechanisms against IFA are proposed which implements detection and reaction approach, similarly, in an independent or collaborative manner. In independent systems, the detection of attack is largely based on network traffic analysis and(or) PIT usage, while the subsequent reaction mechanisms reduces the incoming/outgoing traffic, independently on each router. For instance, Compagno [49] and Gasti et al. [87] proposed to detect the IFA through PIT and interest satisfaction analysis (which indeed takes a while to detect). Subsequently, the mitigation is performed by reducing the data rate of incoming interfaces. Vassilakis et al. [201] also proposed a mechanism following a similar manner where attack is detected through anomalous behavior of the consumers, and later the requesting rate of detected nodes is being reduced (or blocked). In collaborative versions of above-mentioned mechanisms, the intermediate routers also exchange the attack information with each other. Although this enhances the efficiency of mechanisms, however, generates additional signalling overhead. Such as [49, 60, 201] also issues push-back alert messages to the downstream interfaces to reduce the data rate.

The authors in [60] also proposed a collaborative countermeasure knows as Interest Traceback. In particular, the detection is triggered on the basis of PIT size increase, and produces artificial spoofed data packets for each interest stored in PIT. Eventually, during attack, the data packets traces the interests generators. The limitation of the approach includes excessive amount of additional traffic generation in the network, resulting in the deple-

tion of bandwidth and performance. In particular, all the countermeasures tries to limit the number of overall incoming interests (i.e., including benign and malicious), either at each interface [49], [19] or router [60]. Therefore, results in performance degradation of legitimate users and requires further enhancements in terms of traffic differentiation between benign and malicious traffic.

5.2.2 Mitigating DDoS with a stateless Active Queue Management schemes in IP

The congestion handling techniques, such as Active Queue Management (AQM) have gathered significant attention from research community to mitigate DoS attacks in existing Internet protocol [34,92,103,217]. AQM methods are mainly classified in two immense categories based on their functionality and considering the type of traffic they are able to handle. The first category aims to provide fairness during network congestion where the incoming traffic consists of only responsive flows, such as RED [81], BLUE [79], and AVQ [117]. However, the second category functions to provide fairness when the incoming traffic consist of both responsive and unresponsive flows, such as CHOKe [160], SFB [78], and FRED [134].

Random Early Detection (RED) tries to acquire a better queue stability by estimating the level of congestion in the router's buffer and drops packets accordingly using an exponentially weighted moving average (EWMA) of the queue length. One of the RED's limitations is that it is unable to identify unresponsive flows, therefore, requires significant parameters tuning to attain optimal results. Several methods have been proposed to address this limitation which are based on the idea of (or function with) RED. Examples includes CHOKe [160], xCHOKe [47], and RECHOKe [217] and FRED [134]. In precise, CHOKe is a stateless technique which tries to handle unresponsive flows by identifying them and then penalize them by dropping more of their packets.

5.3 Mitigation of IFA exploiting AQM

In this chapter, we take a footstep in the direction of identifying and differentiating malicious packets from the benign traffic during IFA. In parallel, aiming to maintain the simplicity of interest forwarding. In particular, we exploit the idea of active queue management algorithm [160] with aim of stabilizing the router's PIT. The proposed protocol follows the phenomena, i.e., CHOose to Kill malicious Interest, CHOose to keep genuine Interest for

IFA (ChoKIFA), and targets to provide fairness among the benign interest packets that pass through the router. ChoKIFA differentially penalizes malicious traffic by dropping more of its interests packets, without using any global knowledge of the network (i.e., state information).

Approach overview

The fundamental idea behind ChoKIFA is to exploit the PIT state which forms adequate statistics regarding the incoming and outgoing interest packets, and use it to identify and drop malicious interest packets. When an interest arrives at the router, ChoKIFA randomly draws an interest from the PIT and compare it with the incoming interest. If both the interests belongs to the same traffic flow[1], then both are dropped, otherwise the randomly drawn interest is left stored in the PIT. Moreover, the compared incoming interest is stored in the PIT with a probability that depends on the level of PIT occupancy. The intuition of ChoKIFA is that, during IFA, the PIT of a router is likely to have more entries belonging to malicious interests[2], therefore, malicious interests are more likely to be chosen for comparison. Categorically, interests belonging to malicious flows arrives numerously and are more probable to cause comparisons. The intersection of these two high probability events grows higher during attack, therefore, the interests belonging to malicious flows are being dropped.

5.3.1 System Model

In this work, we consider the topology illustrated in Figure 5.1, as used by various authors [19, 60]. Multiple benign consumers (let say, C) retrieving the desired content from a producer (P) which is publishing the content under specific name prefix ($prefix$). We focus on the case where C requests existing content, i.e., the interest which is being satisfied by the P, such as $prefix/data$. We call these interests Benign Interests (BIs). BIs and the corresponding content packets traverse multiple routers (R) before being satisfied by P. In our scenario, each router $r_i^j \; \epsilon \; |R|$ has the default settings of ICN [221], where j is the interface of i-th router. In addition, each router performs caching and uses *best route* as the forwarding strategy.

[1]The traffic flow measurement in ICN is in contrast of Internet Protocol. We present the novel ICN traffic flow comparison in Section 5.3.3.

[2]Recall that unsatisfiable interests points towards the non-existing content and saturates the PIT.

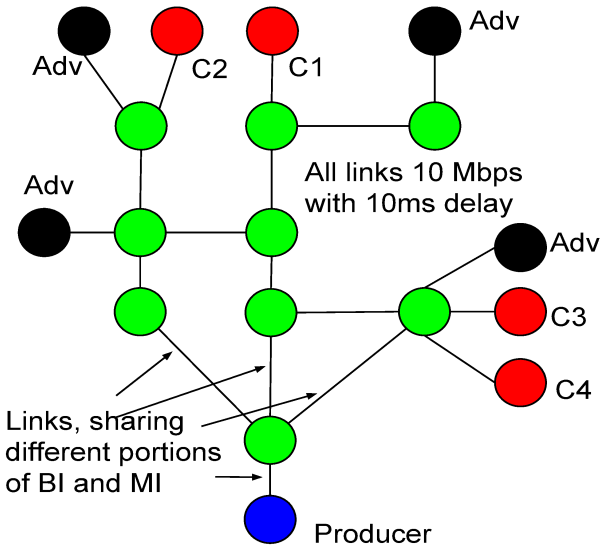

Figure 5.1: Topology considered

5.3.2 Adversary Model

The adversary model of our work is the type III attack (see Section 5.1) where *Adv* generates massive amount Malicious Interests (MIs) which have bogus names to request non-existing content. We assume that *Adv* aims to saturates R's PIT, in particular, by rapid generation of large numbers of MIs [19, 49, 205]. Once the PIT is completely full, incoming BIs are being dropped. This has more than a few consequences. First, MIs referring to non-existing content remains stored in the $R's$ PIT. Second, the sending rate of MIs is not dependent on the bandwidth allocated by the R to content packets, or on the capability of *Adv* to receive content. Third, MIs cannot be replied back by the $R's$ caches. Lastly, if created sophisticatedly (e.g., with a random component at the end of each name) MIs are never collapsed until the interests decays. All these effects allows *Adv* to efficiently fill up $R's$ PIT, which makes the attack more damaging than type I and type II IFA.

Without the loss of generality, we assume that *Adv* is capable to corrupt set of C (i.e., botnet), through which it triggers the attack. This assumption is realistic and justified by the current scenarios of DDoS attacks [86]. *Adv* issues the MIs under the name $prefix$ which is registered by P. In order to make each MI reach towards P, *Adv* attaches a random value with each interest, i.e., $prefix/Rnd$ where Rnd is a random string. Moreover, the percentage of bots is taken 50% with the ratio of C in the whole network [19].

Lastly, similar to C, Adv starts sending MIs at time t. Table 5.1 summarize the notations used in the chapter.

Table 5.1: Summary of notations used

Notation	Meaning
Adv	adversary
C	consumer
P	producer
R	set of all routers
MI	malicious interest
BI	benign interest
r_i	i-th router, $r\epsilon\|R\|$
r_i^j	j-th interface of router i
$\delta(r_i^j)$	interest satisfaction ratio
$\delta_{th}(r_i^j)$	threshold for interest satisfaction ratio
$\rho_{avg}(r_i^j)$	average PIT size
w_ρ	weight factor for EWMA
$\rho_{size}(r_i^j)$	actual PIT size
$\rho_{th}^{min}(r_i^j)$	minimum threshold for PIT size
$\rho_{th}^{max}(r_i^j)$	maximum threshold for PIT size
P_b	interest drop probability
P_{max}	maximum drop probability

5.3.3 ChoKIFA: CHOose to Kill Interest Flooding Attack

In this section, we present the details our proposed mitigation mechanism for IFA. In order to be effective in defending against IFA, the mitigation approach has to be able to detect and differentiate malicious requests from benign ones. In this regard, ChoKIFA exploits traffic flow as an attribute to differentiate and penalize the MIs from BIs.

Unlike IP, where traffic flow measurement relates to the accountable attributes such as source/destination address, interface number, packets/bytes counts forwarded (source to destination), backward (destination to source) counts and so on [38]. In ICN, following content oriented communication model, traffic flow is centered around series of packets that corresponds to specific piece of data [159]. In particular, traffic flow caries the name of packet that uniquely identifies the content, in accumulation to some further information to transport the multiple chunks or segments which compose the content. Considering this, we design the three novel attributes to compare incoming traffic flow at each router: (i) name-prefix match, (ii) interface

match, and (iii) level of interest satisfaction ratio, i.e., rate between incoming interests to outgoing content, denoted as $\delta(r_i^j)$. The parameter $\delta(r_i^j)$ is an appropriate representation to measure routers capability to satisfy interest on a particular interface [21, 49]. In particular, $\delta(r_i^j) > 1$ denotes that the number of content packets received at router r_i^j is less than the number of interests forwarded from the same interface.

In order to mitigate IFA, ChoKIFA dynamically computes the actual size of the PIT, denoted as $\rho_{size}(r_i^j)$, at each instance. Further, ChoKIFA marks two thresholds on the PIT size, a minimum threshold ($\rho_{th}^{min}(r_i^j)$) and a maximum threshold ($\rho_{th}^{max}(r_i^j)$), as well as, a threshold for interest satisfaction ratio, denoted as $\delta_{th}(r_i^j)$ [49]. For each interest arriving at r_i^j, if the PIT size is less than the $\rho_{th}^{min}(r_i^j)$, the interest gets stored in the router's PIT. If all the interests requested by C are satisfied by P or router's cache (i.e., no interest requested for non existing content), then PIT size should not reach up to $\rho_{th}^{min}(r_i^j)$, frequently.

In case of IFA, *Adv* requests massive amount of MIs to the network, therefore, PIT size will be influenced and exceeds the normal range. When the PIT size is greater than $\rho_{th}^{min}(r_i^j)$ and less than $\rho_{th}^{max}(r_i^j)$ (i.e., $\rho_{th}^{min}(r_i^j) < \rho_{size}(r_i^j) < \rho_{th}^{max}(r_i^j)$), each new incoming interest is compared with the randomly selected interest from PIT, named as *drop interest candidate*. If both the interests have the same traffic flow then both are dropped. This choice is motivated by the fact that all the entries in PIT are likely to be occupied by MIs (i.e., under IFA). Recall that during attack, interest flooding saturates the PIT with MIs, pointing towards no existing content. The key attributes to identify the traffic flow of each new incoming interest are three subsequent conditions: (i) if it holds the same prefix as of drop interest candidate, (ii) if it is coming from the same incoming interface as of drop interest candidate, and (iii) if both the above conditions holds true, then router compares if the current $\delta(r_i^j)$ exceeds $\delta_{th}(r_i^j)$. On the other side, when the PIT size is greater than $\rho_{th}^{max}(r_i^j)$, all the new incoming interest are being dropped. This leads the PIT occupancy back to below $\rho_{th}^{max}(r_i^j)$.

In contrast, if the new incoming interest is not having the same traffic flow as of drop interest candidate then the randomly selected interest is remained stored in PIT, and the incoming interest is dropped with the probability (P_b) which depends on the average PIT size ($\rho_{avg}(r_i^j)$), as illustrated in Equation 5.1 [81].

$$P_b = \frac{P_{max} * (\rho_{avg}(r_i^j) \text{-} \rho_{th}^{max}(r_i^j))}{(\rho_{th}^{max}(r_i^j) \text{-} \rho_{th}^{min}(r_i^j))}, \tag{5.1}$$

here P_{max} denotes the maximum probability[3]. As the average PIT size varies from $\rho_{th}^{min}(r_i^j)$ to $\rho_{th}^{max}(r_i^j)$, the interest dropping probability P_b varies from 0 to P_{max}. In particular, this means that interest is dropped with the probability of 1 if it arrives when average PIT size exceeds $\rho_{th}^{max}(r_i^j)$, otherwise it is accepted and stored in PIT. In particular, the interest dropping probability is computed by exploiting the mechanism of packet dropping probability of Random Early Detection (RED) [81].

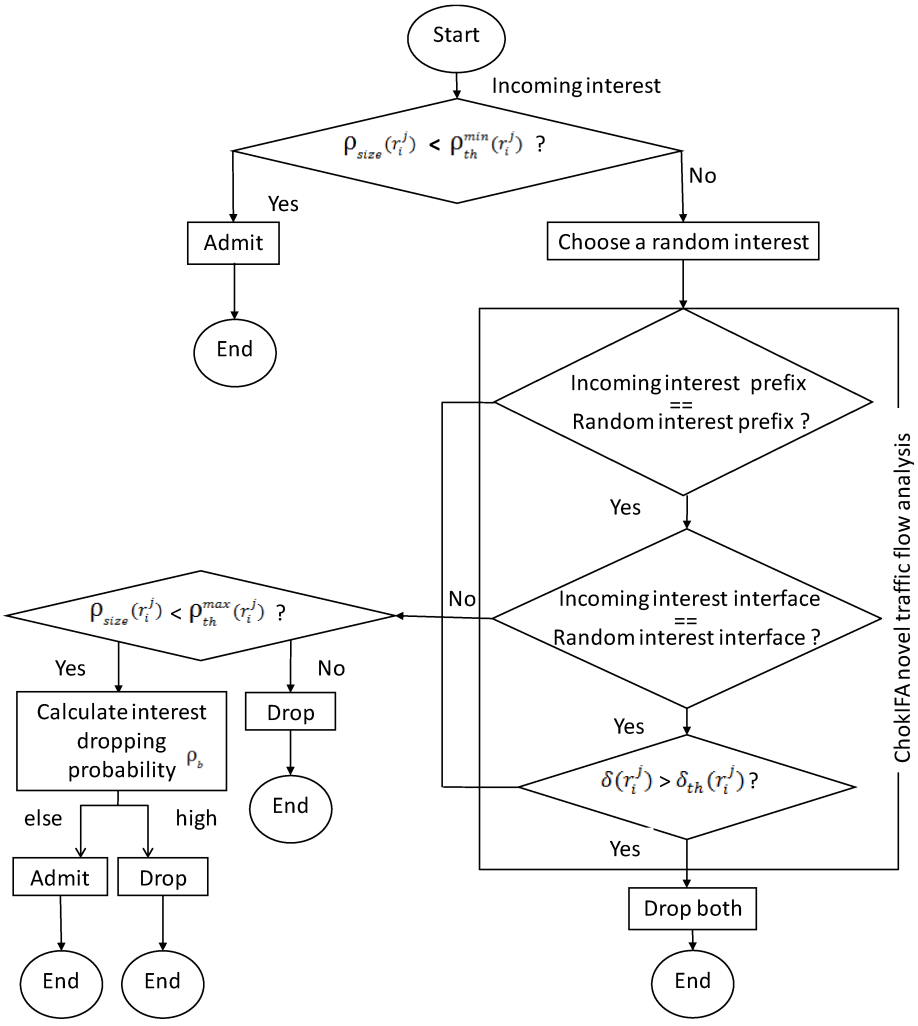

Figure 5.2: ChoKIFA algorithm flowchart

ChoKIFA is a justified stateless algorithm since it does not requires any specific data synchronization between intermediate routers and traffic analyzers. Compared to previous rate limiting IFA mitigating approaches [19],

[3]We take the value of maximum probability (P_{max}) to be one.

it requires few operations, i.e., draw a random interest packets from PIT, compare traffic flow and possibly drop both the incoming and drop interest candidate from PIT. Hence comparing to previous mitigation approaches, ChoKIFA not only restricts the rate of new incoming MIs, but also remove MIs which are already stored in PIT during attack. A detailed flow chart of ChoKIFA is given in Figure 5.2.

5.3.4 Parameters setting

In our proposed approach, the parameters, $\rho_{avg}(r_i^j)$, $\rho_{th}^{min}(r_i^j)$ and $\rho_{th}^{max}(r_i^j)$ are essential so that network designer makes sensible decisions about the desired average PIT size which directly impacts the interest dropping probability. This ensures to handle small bursts of benign traffic which may encounter default network congestion or content retrieval delay. Below we illustrate few rules for parameters which give effective performance for ChoKIFA under variety of traffic conditions while mitigating the attack.

Ensure adequate calculation of the average PIT size: ChoKIFA calculates the average PIT size ($\rho_{avg}(r_i^j)$) using an exponential weighted moving average (EWMA). The use of EWMA for calculating $\rho_{avg}(r_i^j)$ makes sure that the short term increase in PIT size which may result from a burst of benign incoming interests (e.g., which are not satisfied due to network congestion/delay from the producer) do not result in the significant increase of average PIT size.

Equation 5.2 illustrates the calculation of the $\rho_{avg}(r_i^j)$ where w_ρ is the weight factor for calculating EWMA and $\rho_{size}(r_i^j)$ is the current/actual PIT size. Then the average PIT size used for interest dropping probability is updated as [81],

$$\rho_{avg}(r_i^j) = (1 - w_\rho) * \rho_{avg}(r_i^j) + w_\rho * \rho(r_i^j). \tag{5.2}$$

Note that the calculation of average PIT size can be made particularly efficient when w_ρ is set as a negative power of two. If w_ρ is too large, then the averaging procedure will not filter out the temporary congestion of PIT [81].

Setting a minimum threshold for the PIT size: The optimal values of $\rho_{th}^{min}(r_i^j)$ depends on the desired level of average PIT size and default network conditions. In case, the typical traffic is fairly bursty and congested, then the $\rho_{th}^{min}(r_i^j)$ should be correspondingly large to allow PIT utilization to be maintained at an acceptably high level.

Setting $\rho_{th}^{max}(r_i^j) - \rho_{th}^{min}(r_i^j)$ sufficiently large to avoid global synchronization: The optimal value of $\rho_{th}^{max}(r_i^j)$ depends in the part of maximum

average delay that can be allowed to interest (e.g., round trip time for interest to retrieve data) and total size of PIT. A useful rule of thumb ChoKIFA implements is to set $\rho_{th}^{max}(r_i^j)$ more than thrice of $\rho_{th}^{min}(r_i^j)$ [81], since the mitigation mechanism works efficiently when max-min is larger than the typical increase in average PIT size.

5.4 Evaluation

In this section, we evaluate the effectiveness of our proposed approach in the presence of IFA. We also compare the performance of ChoKIFA with state of the art IFA mitigation approaches [19] which implements interest rate limiting based on simple limit, interface fairness, satisfaction ratio and limit announcement technique. To this end, we perform extensive simulations using the open-source ndnSIM [20] simulator.[4] We perform the evaluation into two parts, the first part evaluates the harmful consequences of the IFA attack. The second part evaluates the ChoKIFA performance against the IFA and compares ChoKIFA's efficiency with existing countermeasures [19].

5.4.1 Test setup

To set up the tests, we ran our simulations on two different network topologies: a tree topology [60] (as shown in Figure 5.1) and a much larger ISP-like topology (as shown in Figure 5.8). We use tree topology as it represents one of the worst case to defend IFA DDoS attack [19]. While the larger ISP-like topology reflects the performance of mitigation approach when deployed on the real Internet. We implement the topology with a single P and number of consumers, including four honest clients (C) and four adversaries (Adv) connected with multiple ICN routers. Adv requests for non-existing content (MI), which exhibits distinct suffix ($/good/rnd$) compared to valid content ($/good/data$). C requests the interests (BI) for valid content which are entitled to P. Other network parameters and their values used in our test setup are mentioned in Table 5.2.

5.4.2 Evaluation metrics

We evaluate the impact of IFA against ChoKIFA over three metrics which have been widely used in the related work [49, 172, 205]. First, PIT usage which indicates the available capacity of the routers to process benign traffic during an attack. Second, the percentage of BIs and MIs dropped by

[4] ndnSIM implements the NDN protocol stack on NS-3 simulator.

Table 5.2: Parameters for simulation

Parameters	Value
Interest sending rate for C (interests/second)	30
Interest sending rate for Adv (interests/second)	1000
Interest size (kilobyte)	1
Number of C	8
Number of P	1
Number of malicious nodes	4
Number of benign nodes	4
Number of routers	9
Link capacity (Mbps)	10
Link delay (ms)	10
Interest life time (sec)	1
R Total PIT size (kilobyte)	600
ChoKIFA Min threshold for PIT (kilobyte)	1/8 of PIT size, i.e., 75
ChoKIFA Max threshold for PIT (kilobyte)	3/4 of PIT size, i.e., 450
Weight factor (w_ρ) for EWMA	0.001
P_{max}	1
Interest satisfaction ratio threshold	3
Simulation time (sec)	100
Simulator version	ndnSIM version 2.1
Operating system	Ubuntu 16.04

the network during IFA and with proposed countermeasure. This gives an indication of the attack impact and effectiveness of the countermeasure, respectively. Third, we compare the efficiency our proposed approach with existing mitigation approaches in terms of Interest Satisfaction Ratio (ISR) of benign users which is intended to measure the benign traffic received by users during IFA. Precisely, the lower the ISR refer, the greater amount of false positives made by the mitigation approach while distinguishing between the MIs and BIs.

5.4.3 Small-scale simulation

In this section, we present the results of the tree topology to evaluate the impact of attack and effectiveness of ChoKIFA.

5.4.3.1 Interest Flooding Attack impact

Figure 5.3 shows the PIT usage (y-axis) as a function of the simulation time (x-axis). The maximum value on the y-axis corresponds to the total space

available in the PITs (600 kilobytes). The result reports PIT usage before the attack takes place and while the attack is ongoing. No countermeasure is adopted in this case. In particular, the vertical lines (at 20 sec) indicates the instance when adversaries (*Adv*) launch the attack (i.e., start issuing MIs).

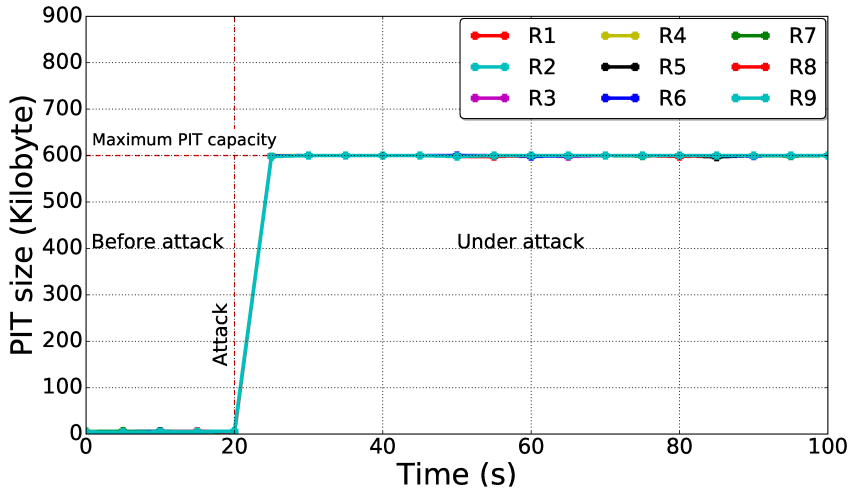

Figure 5.3: PIT usage under IFA without ChoKIFA

Before the 20th second, only legitimate consumers sends benign interests with the rate of 30 interests/second. Since these interests are requested for existing data, therefore, nearly all interests are being satisfied. This scenario keeps the average PIT utilization extremely low and does not cause it to overflow. After adversarial traffic is generated (i.e., after 20th second), which is at the rate of 1000 interests/second, routers are not able to collapse these MIs from the content received or via cached copies, resulting in PIT exhaustion at all the routers (i.e., reach the total PIT size).

Figure 5.4 reports the percentage of total BIs dropped over total received at each router, before and under attack, respectively. Before IFA, only BIs are being requested for existing data, therefore, all BIs are being satisfied by P and none of them is dropped. Under IFA, *Adv* saturates all the routers PITs due to rapid generation of large numbers of MIs. Once the PIT is completely full, subsequent incoming BIs are being dropped. Our results shows (see Figure 5.4) that edge routers attached to benign consumers are highly affected by the harmful consequences of IFA and drops almost 90% of the legitimate traffic. While the next hop routers (i.e., R3, R6, R8) receiving

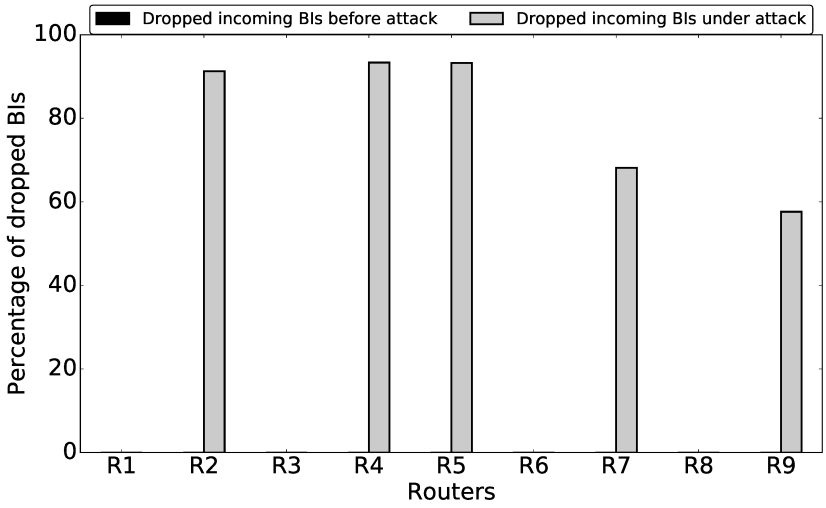

Figure 5.4: Benign interests dropped before and under IFA without ChoK-IFA

both legitimate and malicious traffic on a single interface forwards all BIs since very few remaining BIs are being received by them.

5.4.3.2 ChoKIFA effectiveness and comparison

Figure 5.5 reports PIT usage of all the routers as a function of the simulation time when the proposed countermeasure is active. It must be noticed that, in our simulations to evaluate and compare ChoKIFA under IFA (see Figure 5.5), adversaries launches the attack at different time, i.e., starting from the 20th second, while the benign users starts to request for existing content from the beginning.

Because of the design of CHoKIFA, i.e., minimum threshold of PIT size, approach allows to fill the PIT of all the routers till 75 kilobyte before being able to start traffic flow comparison. In particular, before the minimum threshold of PIT size have not exceeded, it does not start dropping the MIs. Due to this reason, IFA is able to fill PITs of all the routers till 75 kilobytes, as shown in Figure 5.5. In contrast, after exceeding the minimum threshold, ChoKIFA's traffic flow comparison and interest dropping probability does not allow PITs to exceed certain level (i.e., slightly higher than 75) which depends on the dropping probability related to average PIT size. In particular, ChoKIFA then treats each new incoming interest separately by first identifying its traffic flow (i.e., malicious or benign), and then takes the decision to drop MIs which is related to the EWMA average of PIT. Re-

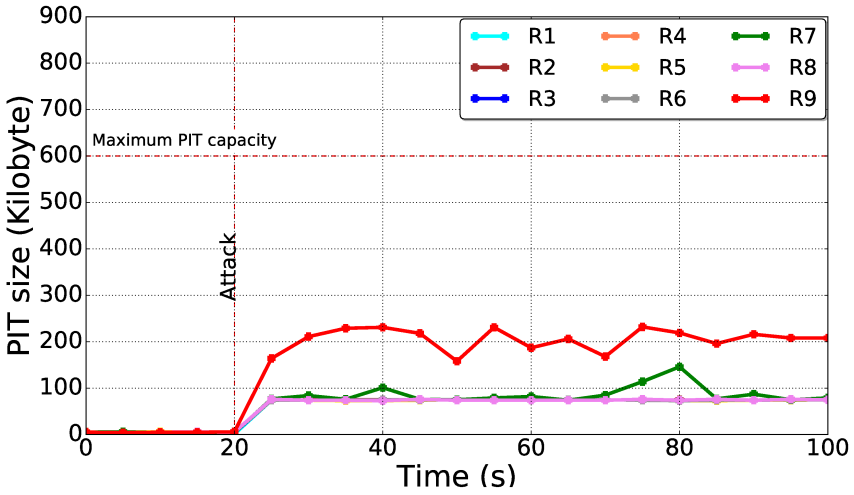

Figure 5.5: PIT usage under IFA with ChoKIFA

sults shows (see Figure 5.5) that gateway node to producer attains slightly higher PIT size than the rest of routers since it receives aggregated amount of malicious traffic from the whole network.

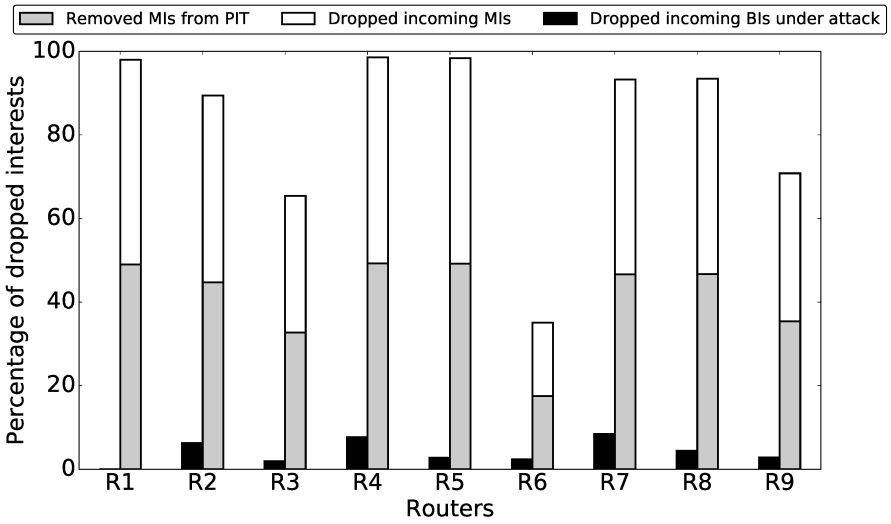

Figure 5.6: Benign and malicious interests dropped under attack with ChoKIFA

Figure 5.6 reports performance of ChoKIFA under IFA, in terms of legitimate and malicious traffic drop. It shows the percentage of total BIs and MIs dropped over total received at each router, respectively. The results

shows the effectiveness of ChoKIFA against the attack. In particular, the legitimate traffic is slightly affected (only 4% of BIs are dropped on an average). Because the PIT is filled up with MIs, therefore, the drawn random interest from PIT is also MI with the very high probability, and in consequence ChoKIFA drops only MIs, i.e., both incoming and already stored in the PIT.

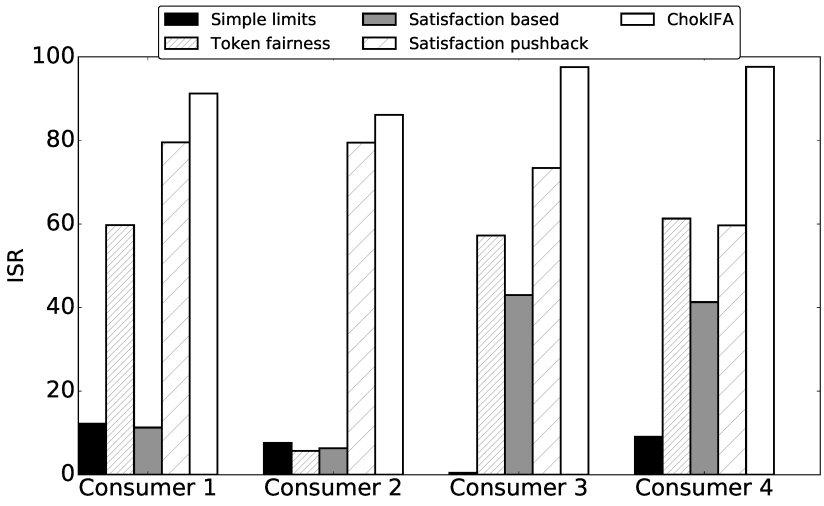

Figure 5.7: Benign consumers ISR comparison

Figure 5.7 reports the ISR of benign users which can be achieved when enabling ChoKIFA. We also compare these results with four different mitigation approaches[5], named as: (i) Simple limit, (ii) Token Bucket fairness, (iii), Satisfaction based acceptance, and (iv) Satisfaction-based pushback. The first three approaches are lightweight and stateless nevertheless not effective in legitimate ISR. Results shows (see Figure 5.7) that Satisfaction-based pushback is slightly effective than previous methods but also induces unnecessary signaling overhead by sending rate limiting announcements continuously in the whole network.

Figure 5.7 reports that ChoKIFA outperforms all four approach in terms of all benign users ISR, remarkably. In particular, during the attack, existing approaches merely limits the overall incoming rate of the interfaces, therefore, drops BIs as well, while ChoKIFA mitigates the attack by differentially dropping MIs and only slighlty affecting BIs. Our proposed defense mechanism is able to main 97% of all benign users ISR. Moreover, it in-

[5]As discussed in Section 5.2, all four approaches are proposed and evaluated in [19], and available on https://github.com/cawka/ndnSIM-ddos-interest-flooding

duces 20 to 60% less false positives comparing to all four approaches while mitigating the attack.

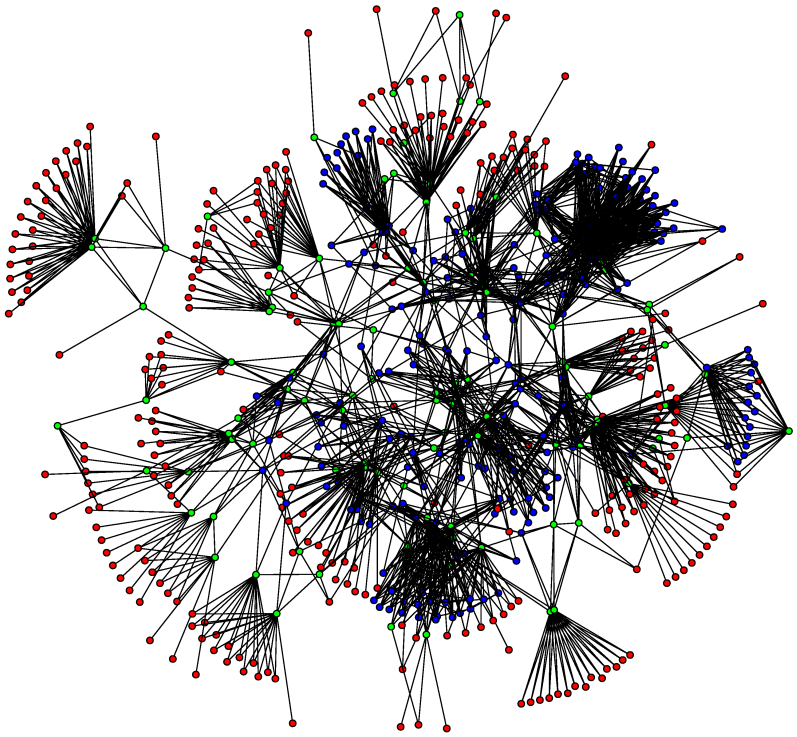

Figure 5.8: Rocketfuel's AT & T topology (AS 7018)

5.4.4 Large-scale simulation

In this section, we evaluate the performance of ChoKIFA by implementing a real ISP-like topology (i.e., AS 7018) which is measured by the Rocket fuel project [189] (see Figure 5.8). There are 625 nodes involved in the topology, which are separated into three categories: clients, gateways, and backbones. The 296 nodes are classified as clients for having degree less than four, while 108 nodes that directly connects to clients are classified as gateways. And the remaining 221 nodes are classified as backbones. The larger ISP topology reflects how our mitigation methods would perform when deployed on the real Internet.

To study the performance of our proposed mitigation strategy under a range of conditions, we varied the percentage of adversary in the network and the frequency with which adversary is sending malicious interests.

5.4.4.1 ChoKIFA effectiveness and comparison

Figure 5.9 confirms that rate limiting approaches [19] are not able to maintain acceptable ISR for benign users in bigger topology as well. In particular, the result shows the percentage of global ISR of all legitimate interests generated in the network, where ChoKIFA maintains almost 97% of ISR during the attack. Note that to show a clear comparison among various mitigation mechanisms, we set the attack duration in this case from 20 to 80 seconds during simulation.

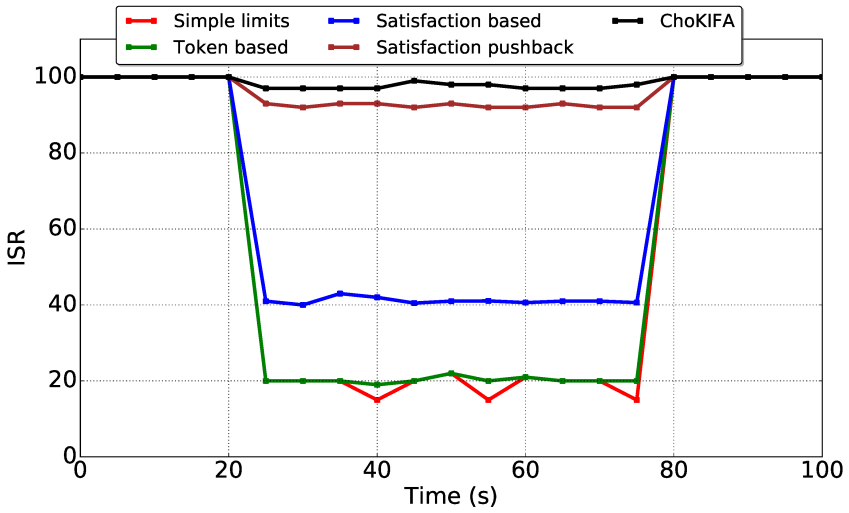

Figure 5.9: Global benign ISR.

Figure 5.10 shows the ISR percentage of total legitimate interests generated in the network when we varied the percentage of attackers in the network - the values ranged from 6% attackers to over 50% attackers in the network. The results are as expected for ChoKIFA and all three state of the art mitigation algorithms. As the number of attackers in the network increases, the lower is the ISR ratio for legitimate interests (i.e., aggregated for all benign users). For instance, in the case of token bucket with per interface fairness algorithm, only 3 attackers can halve the quality of service for the remaining 13 legitimate users. While the two intelligent attack mitigation algorithms also shows a decline in legitimate service quality as the percentage of attackers increases. However, ChoKIFA outperforms all mitigation algorithms and shows a very minor reduction in ISR ratio (i.e., approximately 3%) even when the attacker's percentage is raised more than 50%. Figure 5.11 shows the aggregated legitimate ISR ratio when we increased malicious interest sending rate from 100 interests/second to 10000 interests/second.

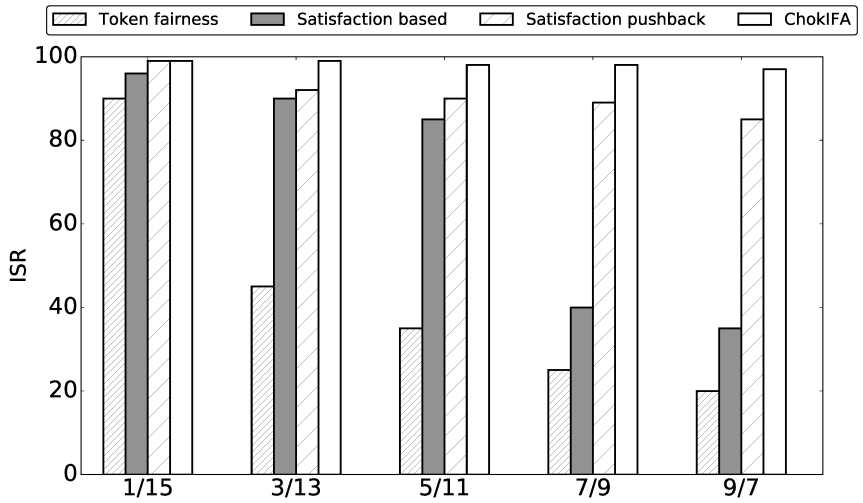

Figure 5.10: Global benign ISR with increasing number of attackers.

The result shows that ChoKIFA remains almost unaffected even with huge amount of increase in malicious interest frequency, while among all state of the art approaches only Satisfaction-based pushback shows satisfactory results.

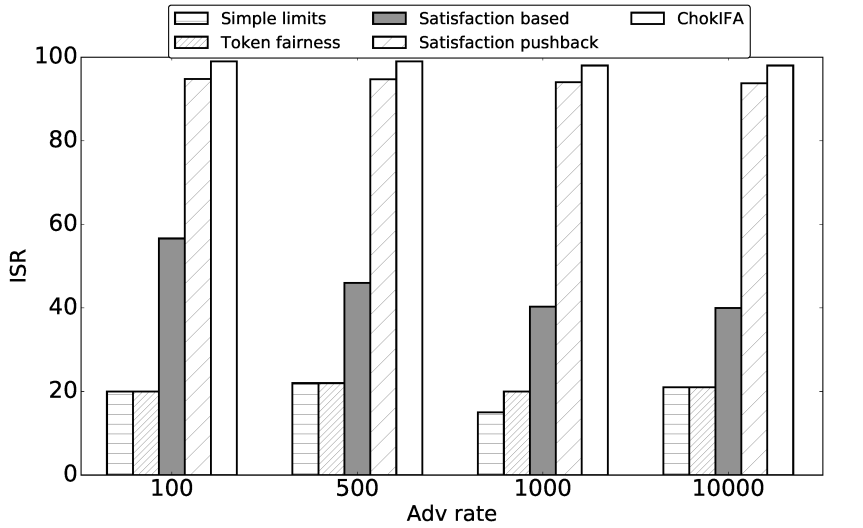

Figure 5.11: Global benign ISR with increasing malicious interest sending rate.

5.5 Summary

In this chapter, we address the interest flooding-based DDoS over ICN, which is explicitly named as IFA. More specifically, we have found that several proposed countermeasures, that adopt detection and reaction mechanisms based on interest rate limiting, are not highly effective and also damage the legitimate traffic.

In our solution, we exploited an active queue management scheme to propose an efficient detection and mitigation mechanism against IFA, which stabilizes the router PIT. The proposed approach penalizes the unresponsive flows generated by adversarial traffic by dropping malicious interests generated during the IFA. To evaluate the effectiveness of our solution, we implemented the proposed protocol on the open-source ndnSIM simulator and compared it with the state-of-the-art. The results report that our proposed protocol effectively mitigates the adverse effects of IFA and shows significantly less false positives in comparison to the state-of-the-art IFA mitigation approaches.

Chapter 6

A Survey on ICN-IP Coexistence Solutions

The decoupling between request sending and content transferring in ICN introduces several benefits, e.g., reduction of latency and network load due to in-network caching [66], inherent content integrity [84], and better support for mobility due to name-based routing [28,76]. However, those benefits can occur only in a full-ICN scenario, which implies a complete replacement of the current Internet. Despite its obvious need, this is a long and complex process which requires the coordination among the different parties (i.e., ISPs) on various attributes, e.g., time, costs for updating hardware and software of the network components and ability to face all the new possible challenges.

Previous attempts to replace a widely used technology, protocol or architecture (e.g., IPv4/IPv6 protocol, 3G/4G technology, 4G/5G technology) have always faced a long period of coexistence between the old and the new solution. In the same way, the replacement of the current Internet will involve a transition phase during which IP and ICN architectures will coexist. More specifically, we envision that in a coexistence scenario there will be ICN and IP "islands" surrounded by an IP or an ICN "ocean", where an "island" will be a single device, a computer, an application or a server running either the ICN or the IP protocol, while an "ocean" will be a network containing components, that run different architectures.

Researchers working in this field have already addressed the coexistence of IP and ICN following two separate approaches. In the first one, the

research groups designed future Internet architectures facing the coexistence only during the deployment of their testbeds and without considering it as part of the initial design. On the contrary, in the second case, the design of the future Internet architectures specifically addressed the coexistence of TCP/IP and ICN.

All the existing architectures are affected by a strong limitation: the lack of a comprehensive approach in addressing the coexistence. The purpose of all those architectures is to improve a network performance indicator, without considering all the issues that arise in a coexistence scenarios, especially the ones regarding the security and privacy of the final users. To design the first complete coexistence architecture, it is necessary first to have a comprehensive overview of strengthens and weaknesses of the existing solutions. Thus, the purpose of this chapter is to provide the first complete analysis and classification of the existing coexistence solutions.

To the best of our knowledge, the report provided by researchers of InterDigital Inc. and Huawei in [168] is the only studies of the existing coexistence architectures available so far. However, it focuses mostly on the different deployment approaches chosen by each solution, while the purpose of our work is to deeply analyze each architecture considering several criteria. Overall, the contributions of this chapter are the following ones:

- We define a set of relevant features which are necessary to comprehensively analyze a coexistence architecture.
- We provide the first comprehensive classification of all the main coexistence solutions.
- We discuss the open issues and challenges affecting the existing coexistence architectures, by providing possible insights to design a more reliable future Internet architecture.

Organization. Section 6.1 illustrates the background which includes basic concepts of the TCP/IP in comparison with the ICN. Section 6.2 describes the criteria we identified and used for the analysis and classification of the coexistence architectures. In Section 6.3, we deeply illustrate each coexistence architecture and provide the motivation for our classification. In Section 6.4, we discuss the main strengths and limitations of the current coexistence architectures, and providing insights for improving the design of the future Internet. Finally, we summarize the chapter in Section 6.5.

6.1 Background

The purpose of this section is to provide first an overview of the main features belonging to the TCP/IP and the ICN architectures (Section 6.1.1) and, then, an illustration of the emerging technologies (Section 6.1.2).

6.1.1 Comparison between Current and Future Internet Architectures

Originally developed as part of the ARPANET project [152] during the 1960s, the current Internet is now often referred as TCP/IP architecture due to its most well-known protocols - Transmission Control Protocol (TCP) and Internet Protocol (IP). On the contrary, the ICN paradigm was first introduced in the TRIAD project [46] in 2000 and, then, followed by several architectures adhering to its new communication model. Since ICN is just the paradigm, we will refer to the following main implementations to describe the technical features of the future Internet: a data-oriented networking architecture (DONA) [112], a content-centric networking architecture (CCN) [101], a named-data networking architecture (NDN) [222], a publish/subscribe architecture (PURSUIT) [67], and a network of information architecture (NetInf) [61].

Protocol Stack

Both TCP/IP and ICN rely on a layered protocol stack, which is comparable to the Open Systems Interconnection (OSI) Reference Model [228], as shown in Figure 6.1. The TCP/IP stack includes the following four layers.

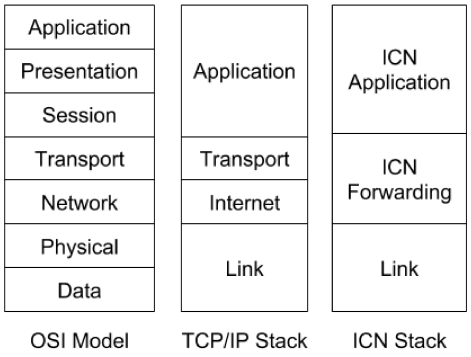

Figure 6.1: TCP/IP and ICN stacks with respect to the OSI layer model

- *Application*: it combines the functionality of the *Application*, *Presentation* and *Session* layers of the OSI model. It is responsible for

sending and receiving data and it is specific for a particular type of application (e.g., Domain Name System (DNS), HyperText Transfer Protocol (HTTP)).

- *Transport*: it targets the *Transport* layer of the OSI model and it is responsible for the end-to-end data transfer and data streams. Its most important protocols are TCP, which provides a reliable and connection-oriented service, and UDP, which offers an unreliable and connection-less service.

- *Internet*: equivalent to the *Network* layer of the OSI model, it provides addressing and routing functionalities to ensure the delivery of messages to their destination. The IP is the most important protocol, but it does not provide flow control or error handling.

- *Link*: equivalent to the *Data* and *Physical* layers of the OSI model, it manages the interaction among physical network components and it works as an interface with the network hardware.

Since the ICN stack is an evolution of the TCP/IP one, each layer is described with respect to the corresponding one in the Internet stack. More specifically, the layers of the ICN stack are the following ones:

- *ICN Application*: the protocols of this layer address content names instead of hosts locations (e.g., the URL inside an HTTP request is replaced with the complete name of a content).

- *ICN Forwarding*: for any ICN-compliant architecture this layer offers the same functionalities as the TCP/IP *Network* layer in such a way that source and destination IP addresses are removed from the network packets and only the addressed content name is declared. According to the specific architecture, this layer can also provide the features of the TCP/IP *Transport* layer. In that case, the interest/data messages replace the TCP/IP segment/ACK messages and the receiver becomes responsible for the message sending rate in place of the sender.

- *Link*: to be ICN-compliant, this layer introduces a mapping between MAC addresses and content names.

Routing

The purpose of the routing functionality is to route network packets from the source node till the destination node on one way and, then, from the destination to the source on the other.

Each TCP/IP packet specifies both the source and the destination nodes by including their IP addresses. An IP address is the unique identifier of each network component and it contains both the address of the network and the address of the specific component within that network. In the current Internet, routers are mainly responsible for the routing functionality. Equipped with at least two IP interfaces (i.e., an incoming and an outgoing one), each router receives IP packets on the incoming interface and checks whether there is a match, based on the longest IP prefix, in its Forwarding Information Base (FIB) internal data structure. The FIB contains a mapping between a network prefix and a router's outgoing interface, together with the next-hop IP address. If there is a match in the FIB for the incoming packet, this is forwarded through the outgoing interface towards the next node in the network.

In the future Internet, the routing functionality differs according to the specific design of each architecture, but they all have a common design choice: the packets sent by a requester contain only the full name of the content, i.e., no IP addresses, neither the requester's one nor the receiver's one. In NDN and CCN architectures, contents are expressed through hierarchical names and routers use a longest name-prefix match approach to find a possible entry in their FIB, which returns the name-prefix(es) towards the next node(es) in the network. On the contrary, DONA exploits a flat naming scheme to point to the contents available in the network and a name-based routing to redirect the request packets until they reach the content producers. A different approach is used by PURSUIT, which relies on a publish/subscribe model. Publishers publish their contents in the network and subscribers ask for a specific content by using a flat naming scheme, made of two components: the rendezvous identifier (RI) and the scope identifier (SI). The first element addresses the component responsible to find the match between publisher and subscriber for a specific content, while the second is used to identify the sub-network where the rendezvous is. Once the subscriber obtains the location of the publisher from the rendezvous node, it sends its packet to the topology manager of the network where the content publisher is. The topology manager, then, identifies the path from the publisher to the subscriber and adds a series of Forwarding Identifiers (FIs) to the header of the packets. After that, the Forwarding Nodes (FNs) forward the packets only using the FIs, without requiring routing tables. Finally, the NetInf architecture adheres to two separate approaches, both name resolution, based on the publish/subscribe paradigm, and name-based routing.

Name Resolution

In TCP/IP architecture there is a dedicated network component responsible for name resolution, which is the Domain Name System (DNS). This is a distributed service, which translates domain names, expressed in hierarchical URLs, into the corresponding IP addresses. The Internet is organized into separate DNS zones, each one under the direct control of an authoritative DNS server, and everytime a network device sends a request to its local DNS server, this might reply with a value saved in its cache or, otherwise, forward the same request to a remote server.

In ICN, the name resolution differs according to the chosen forwarding approach. In case of name-based routing, the requester specifies a content by providing its full name, which is the same analyzed by the routers to find the next hop in the network. On the contrary, in the name resolution approach, like the one used by PURSUIT or NetInf, there is always a dedicated node in the network, which is responsible for the mapping between publishers and subscribers.

Storing

In TCP/IP architecture, the routers are equipped with a buffer memory, which can be used for caching data packets. However, after the first forwarding, routers cannot reuse the same data packets. On the other hand, caching is one of the fundamental features provided by any ICN architecture, where almost any node is able to cache contents and to serve the corresponding requests in the future.

Traffic Management

The current Internet addresses the *Transport* layer protocol through the TCP and the UDP protocols. More specifically, the first one provides a reliable and connection-oriented data transfer between two network components and it is also responsible for the management of the network flow-control. As a matter of fact, before exchanging any data, the TCP protocol checks for the availability of the remote server through its three-way handshake mechanism. Only at the end of the handshake, the real communication starts, together with the data exchange, and it is regulated by the TCP protocol, which introduces sequence numbers in the message blocks to enable the destination node for properly ordering all the received messages.

In ICN, some architectures, such as DONA, still rely on the existing transport protocols so that all the forwarding mechanisms and transport functionalities are guaranteed. However, other ICN solutions, such as NDN, do not provide the *Transport* layer functionalities and, instead, delegate them to the application itself or to the network packets. After a certain

timeout, an application can transmit again a packet, which by design has a limited lifetime to prevent network congestion. Moreover, the availability of distributed caches, which means contents, all over the network should prevent losses due to congestion.

6.1.2 Emerging Technologies

Before thinking of redesigning the whole Internet architecture, researchers and companies have provided several solutions, which work on top of the current Internet, to overcome some of its limitations. Among those, the most successful attempts are the following emerging architectures: Software-Defined Networking (SDN), Network Functions Virtualization (NFV), Content Delivery Network (CDN) and Delay Tolerant Network (DTN).

Software-Defined Networking

SDN [77] is an emerging networking paradigm that separates network control logic (i.e., the control plane) from the underlying switches and routers that forward the traffic (i.e., the data plane). By separating the control and data planes, the network switching/routing devices become simple forwarding devices and the control logic is incorporated in a logically centralized controller. This separation primarily helps in simplifying network (re)configuration, policy enforcement, and evolution [114]. The control plane and the data plane communicate via a well-defined programming interface, i.e., the forwarding elements of the data plane request for instructions from the controller as well as the controller has direct control over the data plane elements using an Application Programming Interface (API). The most popular flavor of such an API is OpenFlow [141]. An OpenFlow switch has one or more flow tables for handling packet-rules. When a rule matches with the incoming traffic, the OpenFlow switch performs certain actions (forwarding, modifying, dropping, etc.) on the traffic flow. The rules installed by the controller decide the role of an OpenFlow switch, i.e., it can behave as a switch, router, firewall, or middlebox (such as traffic-shaper, load-balancer).

Network Functions Virtualization

Diversity and dominance of proprietary appliances made service deployment, as well as testing, complex. NFV [130] was designed as a technology to leverage IT virtualization by exporting network functions from the underlying dedicated hardware equipment to general software running on Commercial Off-The-Shelf (COTS) devices. Using NFV, the key network functions can be performed at various network locations, e.g., network nodes, data-

centers, network edge, as required. NFV is different from SDN, and it only deals with the virtualization of network functions.

Content Delivery Network

The initial implementation of the Internet was designed to manage the traffic in a passive, end-to-end, and "best effort" approach [199]. With the explosion of user data and commercial content over the Internet, the "best effort" approach for traffic management became inefficient and unscalable. To handle this situation, CDN [199] was designed [155,209]. Nowadays, CDN appears as an integral and essential overlay network for the Internet [48,142, 190]. CDN primarily aims to improve bandwidth availability, accessibility, and precise content delivery through content replication.

CDN's architecture consists of several cache servers that are strategically located across the Internet. Typically, CDN holds a hierarchy of servers with multiple Points-of-Presence (PoP) that stores copies of identical content to satisfy user's demand from most appropriate/closest site [156]. It also has back-end servers for intra-CDN content distribution. CDN categorically distribute web contents to the cache servers, which are positioned close to the users. As a result, CDN offers fast, efficient, and reliable web services to the users.

Though CDNs improve content delivery, but their performance is limited by the underlaying ISPs. Usually, CDNs do not manage independent packet data services, rather they rely on the ISPs to make packet routing decisions. Moreover, both ISP and CDN collectively provide end-to-end QoE[1] for content delivery. Thus, coordination between ISP and CDN providers causes a massive impact on the overall QoE [190].

Delay Tolerant Network

DTN is an overlay architecture which operates above the protocol stack of *ad hoc* wireless networks and enables gateway functionality to interconnect them [108]. To provide communication among networks having excessive delays due to highly repetitive link disruptions, DTN exploit various methods such as storage capacity, replication, parallel forwarding, forward error correction, and variety of protocol techniques. Precisely, DTNs implements *store-and-forward* architecture in which data units that are characterized as bundles are temporarily stored at the nodes (during network disruptions) until an appropriate next hop is discovered.

[1] QoE is an all-inclusive model, which defines the perceived quality subjected to the user when retrieving content or applications over the Internet.

6.2 Features and Evaluation Parameters of the Coexistence Architectures

In order to classify the existing architectures, we identified a set of features and evaluation parameters as the necessary ones to have a complete overview of each coexistence solution. The former come with the design of a coexistence architecture, while the latter refer to the challenges introduced during its deployment in a real scenario. The features are the following ones: *deployment approaches*, *deployment scenarios*, *architecture or technology used* and *addressed coexistence requirements*. On the contrary, the evaluation parameters are: *traffic management*, *access control*, *scalability*, *dynamic network management* and *latency*. In the remaining part of this section, we will describe in details first the features (Section 6.2.1) and then the evaluation parameters (Section 6.2.2) used for analyzing each coexistence architecture.

6.2.1 Features

Deployment Approaches

The deployment of ICN into a TCP/IP architecture inevitably rises the following question: *How to introduce the ICN protocol into the TCP/IP one?* To achieve this aim, researchers identified three possible approaches, shown in Figure 6.2: *overlay* in case of ICN running on top of the IP protocol, *underlay* in case of ICN running under the IP protocol and *hybrid* in case of a coexistence of both IP and ICN protocols. In the *overlay* deployment approach, the aim is to enable the communication among several ICN "islands" in an IP "ocean" and is achieved through a tunnel over the current Internet protocol. On the contrary, the *underlay* solution involves the introduction of proxies and protocol conversion gateways near to either ICN or IP "islands" to properly deliver and receive outgoing and incoming requests. As an example, an HTTP request sent to an ICN "island" is intercepted by a gateway, which is responsible for translating it into an ICN Interest. Then, the resulting ICN Data packet is translated again into an HTTP reply sent back to the requester. Finally, the *hybrid* approach claims the coexistence of both ICN and IP, by adopting dual stack nodes which are able to handle the semantics of both IP and ICN packets. Given the diversity of the two protocols, from a semantic and format point of view, a dual stack node can use various options to infer content names from an IP packet, such as performing deep packet inspection in the payload or looking into the content name in the IP option header.

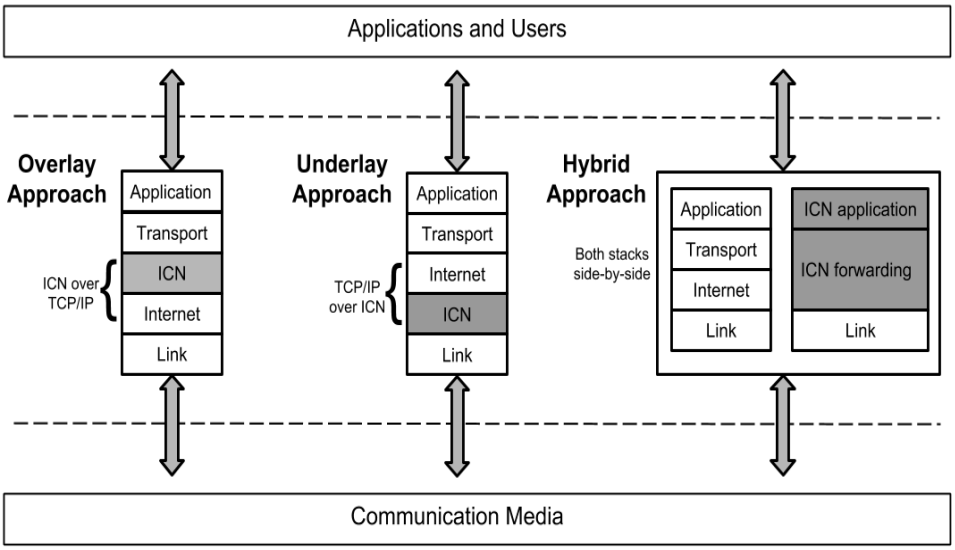

Figure 6.2: Overview of the possible deployment approaches of ICN in a TCP/IP architecture

Deployment Scenarios

The purpose of this feature is to analyze all the possible scenarios in which a coexistence architecture can be deployed among the ones we identified and which are illustrated in Figure 6.3. Each deployment scenario involves two "islands", which run either the same networking architecture or two separate ones, surrounded by an ICN or an IP "ocean". The possible different deployment scenarios are the following ones:

- Communication between two ICN "islands" through an IP "ocean".
- Communication between an ICN "island" and an IP "island" through an IP "ocean".
- Communication between an ICN "island" and an IP "island" through an ICN "ocean".
- Communication between two IP "islands" through an ICN "ocean".
- Communication between different "islands" in separate "oceans" through the *Border Island*.

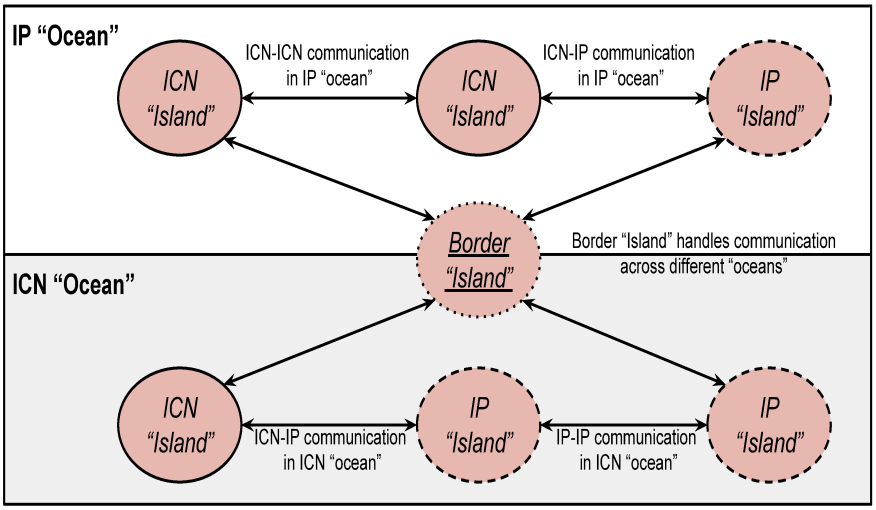

Figure 6.3: Overview of the possible deployment scenarios for a coexistence architecture

Addressed Coexistence Requirements

In a coexistence scenario, the heterogeneity of the different networks might generate conflicts that prevent each individual architecture from guaranteeing its main features and properties. For example, since most of the ICN architectures do not preserve the native transport functionalities provided by the TCP protocol of the current Internet, one of their most significant limitations is the traffic management. In a coexistence scenario, there would be a conflict between an IP "island" implementing its own logic for managing the traffic network and an ICN "island", which does not support the same features. Examining previous works in the literature [171], we consider forwarding, storage, security and management as the necessary requirements to be supported in a coexistence scenario.

Forwarding: The network forwarding devices should be able to handle packets with diverse routing identiers (e.g., the variable-lengths of content names lead to dissimilar size of prefix-set and thus, different forwarding table lookups).

Storage: The network devices should support in-network caching to serve the content request and reduce bandwidth consumption. Nevertheless, the storage capacity of the network devices also affects the size of the index table for the cached content and the time required to match the content's name in the index table.

Security: The network devices should preserve the security policies enforced in one (source) network to another (destination) network such as authenticating the digital signatures of content objects for content-based security or privacy policies.
Management: The network devices should support management-related operations such as traffic-shaping/engineering, load-balancing, and explicit path steering.

Architecture or Technology Used

ICN and IP architectures are not the only ones that can coexist, and even the coexistence could be improved using other technologies. More specifically, ICN well fits with several different technologies that are already deployed in the current Internet infrastructure. Among those, there are SDN, NFV or CDN. The purpose of this feature is to collect all the technologies that the coexistence solutions involve.

6.2.2 Evaluation Parameters

As evaluation parameters, we considered the following challenges arising during the deployment of a coexistence architecture in a real scenario:
Traffic management: Network traffic management is the process of controlling, managing, and reducing network traffic to reduce congestion, latency, and packet loss. Typically, traffic management involves a network scheduler for: (i) traffic shaping, i.e., to the retime (delay) packets until they meet prespecified limits and (ii) traffic policing, i.e., to discard (drop) as well as decrease (demote) priority of packets that exceed some prespecified limit.
Access control: In the context of the networks, access control uses a set of protocols to define, implement, and maintain policies that describe how to secure access to the network nodes by users/devices. Typically, it includes: (i) authorization, authentication, and accounting of network connections, (ii) identity and access management, (iii) Mitigation of non-zero-day attacks, (iv) policy life-cycle management, (v) role-based controls of user/device and application, (vi) and security posture check.
Scalability: Scalability is an attribute that ensures that the overall performance of the network will not affect by the size of the network. In other words, scalability describes the ability of a network to grow and manage increasing demand.
Dynamic network management: It is the process of administering and managing dynamic changes in computer networks, such as topology changes and handovers for seamless host mobility.

Table 6.1: Classification of the available coexistence solutions.

Parameter			PURSUIT	NetInf	NDN & CCN	O-ICN	CONET	DOCTOR	GreenICN	coCONET	POINT	RIFE	CableLabs	NDN-LAN	HICN	OFELIA
Duration of the project/ Year of publication			2010 to 2013	2010 to 2013	2010 till today	2015	2010 to 2013	2014 to 2017	2013	2012	2015 to 2017	2015 to 2018	2016	2017	2018	2012
Design features	Deployment approaches	Overlay	✓	✓	✓	✓	✓		✓	✓						
		Underlay						✓			✓	✓	✓			
		Hybrid					✓							✓	✓	✓
	Addressed coexistence requirements	Forwarding	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Storage	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
		Security	✓	✓	✓			✓		✓	✓	✓			✓	✓
		Management					✓		✓	✓					✓	✓
	Architecture or technology used		PSIRP, LAN			SAIL, SDN		NFV, SDN	SDN	SDN	PURSUIT, SDN	PURSUIT, DTN	CDN	LAN	DNS	CONET, SDN
	Deployment scenarios	ICN-ICN communication in IP "ocean"	✓	✓	✓	✓	✓	✓	✓	✓				✓	✓	
		ICN-IP communication in IP "ocean"						✓	✓	✓			✓	✓	✓	
		ICN-IP communication in ICN "ocean"						✓					✓	✓	✓	
		IP-IP communication in ICN "ocean"						✓					✓	✓	✓	
		Border Island				✓	✓	✓			✓	✓			✓	✓
Evaluation parameters		Traffic management	✗	✗	✗	✗							✗	✗		
		Access control		✗				✗								
		Scalability				✗			✗		✗	✗		✗	✗	
		Dynamic network management				✗					✗	✗		✗		
		Latency									✗	✗		✗	✗	
		Other		transport layer functions.			New IP option overhead.		SDN controller management.	ICN/OpenFlow-compliant network.			HTTP/CCN translation.			OpenFlow-compliant.

✓ Addressed ✗ Not addressed

6.3 Analysis and Classification of Coexistence Architectures

The purpose of this section is to illustrate the analysis and classification of the coexistence architectures according to the features and the evaluation parameters described in Section 6.2. Table 6.1 summarizes the analysis and classification which is presented at the end of the section.

6.3.1 Publish-Subscribe Internet Technologies (PURSUIT)

PURSUIT [196] is a European project financed by the Seventh Framework Program (FP7) started in September 2010 and ended in February 2013. PURSUIT is an evolution of the FP7 project Publish-Subscribe Internet Routing Paradigm (PSIRP) [67], proposing an ICN model based on a source node, that publishes an information, and on a client node, that subscribes to the content it desires. If the information is available, it will be delivered to the client. PURSUIT aims at improving PSIRP, meanwhile evaluating its performance, scalability, and coexistence with the current Internet network.

The architecture proposed in PURSUIT, which is shown in Figure 6.4 in a simplified form, relies on the definition of a new data format and on the introduction of three new network components. PURSUIT addresses the data as information items, which consist of pair of identifiers, i.e., Rendezvous ID (RId) and Scope ID (SId). The former represents the real piece of information, while the latter specifies the group which the information belongs to. The three additional network functions addressed by PURSUIT are: Rendezvous Function, Topology Function, and Forwarding Function. The Rendezvous Function plays a fundamental role in PURSUIT since it maps subscribers to publishers and supports names resolution. Moreover, it also initializes the delivery of information item to the client. The *Rendezvous Node (RN)* performs the Rendezvous Function and relies on the hierarchical Distributed Hash Table internal data structure. The *Topology Manager (TM)* implements the Topology Function by deploying a routing protocol to collect the topology of its domain and by exchanging routing information with other domains for global routing. The *Forwarding Node (FN)* implements the Forwarding Function and it is also responsible for redirecting the information item to the client. In particular, the forwarding mechanism is label-based and uses the Bloom Filter technique [104] to speed up the information delivery. In addition, the *FN* offers also a caching facility.

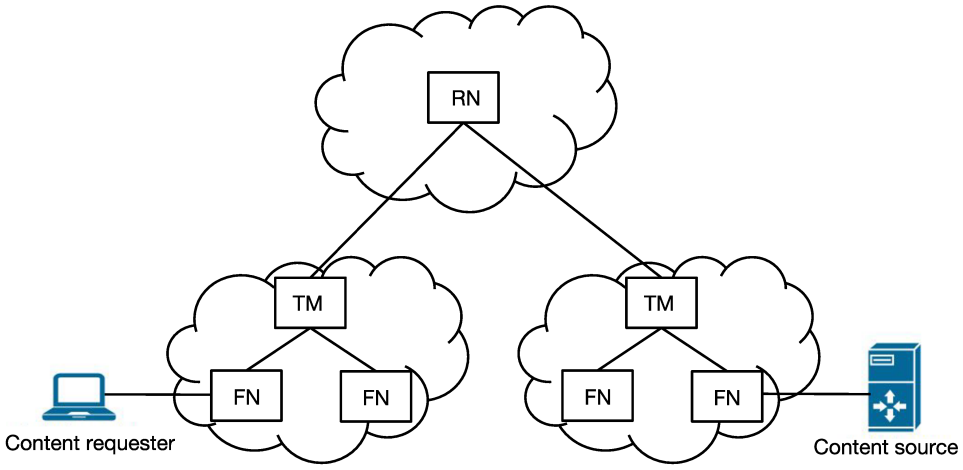

Figure 6.4: Simplified view of the PURSUIT network architecture.

As shown in Figure 6.5, the PURSUIT node internal architecture encompasses several components, enabling the publish/subscribe communication model among the different stack layers. The *IPC element* implements a non-blocking interprocess mechanism, allowing users-space applications to issue publish/subscribe requests and communicate through the proposed prototype. The functionality of the *local proxy* element is to maintain a local record for all the pending subscriptions and, after receiving a request, dispatch it to the appropriate functions (i.e., *Rendevous Function*, *Forwarding Function*, *Topology Function*). Finally, the *communication elements* are responsible for transmitting publications to the network. The design implementation in [196] is based on Click elements [111] which creates Ethernet frames and forwards them to the appropriate network interface. In addition, it provides the ability to utilize raw IP data packets as an alternative mechanism. This enables the prototype to be tested in Internet-wide scenarios.

Deployment Approach: Trossen et al. [196] implemented a Layer-2 VPN-based *overlay* solution of PURSUIT among multiple nodes located in Europe, US and Asia. The prototype is established and verified on three different testbeds for experimental purposes, functioning as an overlay on LAN environment. To showcase a specimen of native ICN application, multimedia streaming services were hosted as a demonstration, showing a lossless transmission and comparable performance.

Deployment scenarios: The most suitable coexistence scenario for deploying PURSUIT involves the communication between two ICN "islands" through an IP "ocean", which is guaranteed by the *overlay* approach adopted in the PURSUIT testbed.

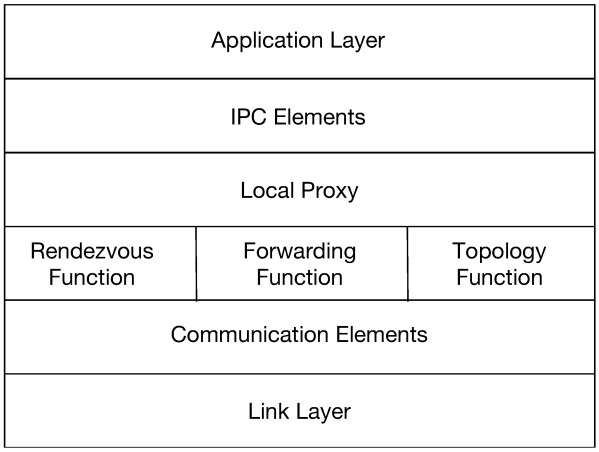

Figure 6.5: Internal architecture of a PURSUIT node in an *overlay* deployment [196].

Addressed coexistence requirements: PURSUIT guarantees the following three coexistence requirements:

- Forwarding: this is specifically provided by the *FN*, a software-based forwarder used for ICN messages exchange;
- Storage: the *FN*, which has the responsibility of redirecting information to the client, provides caching facility to furnish storage of information;
- Security: the security measures provided by PURSUIT refer to the access of information. Besides gathering information into groups, PURSUIT supports the information categorization into scopes, used for the definition of access privileges and policy implementations;

Architecture or technology used: PURSUIT is an evolution of the PSIRP project and its testbed has been realized as an *overlay* solution over a LAN environment.
Evaluation Parameters: The main issue introduced by the *overlay* deployment in the PURSUIT architecture is the traffic management. This is mainly due to the existing Internet applications and protocols, which are not completely compatible with the techniques implementing ICN over TCP/IP or UDP [147, 161, 184, 222] for traffic transport. Thus, many applications and protocols, such as HTTP based multimedia streaming protocols, might face false throughput estimations [54]. This is due to the TCP aggressiveness in presence of variations in content source location (e.g., dynamic caching and interest aggregation) [55].

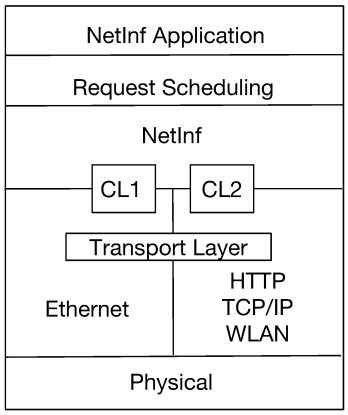

Figure 6.6: Internal architecture of a NetInf node in an *overlay* deployment [61].

6.3.2 Network of Information (NetInf)

The NetInf architecture [61] is the approach proposed by the European FP7 project SAIL [7], started in January 2010 and ended in February 2013. The key components of NetInf architecture are the Convergence Layers (CLs), which are able to map the information, expressed through any protocol (e.g., HTTP, TCP, IP, Ethernet), into specific messages compliant to a general communication paradigm. In particular, when two nodes communicate between each other, the functionality of a CL is to provide framing and message integrity to NetInf *requests* and *responses*.

Figure 6.6 depicts the different *CLs* designed within the NetInf stack. This encompasses an additional function (i.e., *Request Scheduling*) between *NetInf applications* and the *NetInf protocol*. The *CLs* make NetInf able to function over a variety of networks links and protocols such as HTTP, TCP/IP, WLAN and Ethernet. Moreover, the *CLs* also provide transport layer functions across different nodes such as flow control, congestion control and reliability.

Deployment Approach: NetInf adheres to the *overlay* deployment approach, as it is confirmed by its first prototypes, deployed as an *overlay* strategy over TCP/UDP.

Deployment scenarios: NetInf is originally provided as an *overlay* approach, and for this reason it supports the communication between two ICN "islands" through an IP "ocean".

Addressed coexistence requirements: The coexistence requirements provided by NetInf are the following ones:

- Forwarding: NetInf guarantees both name-based forwarding and name resolution; NetInf message forwarding protocol relies on the lower-layer networking technology (e.g., TCP connection between two Internet hosts) and this communication is provided by the CLs;
- Storage: NetInf nodes support both on-path and off-path caching;
- Security: the CLs are responsible for the integrity of the messages exchanged in the architecture;

Architecture or technology used: Besides the standard TCP/UDP/IP tunneling, which is part of the *overlay* approach, NetInf does not rely on additional architectures.
Evaluation Parameters: The deployment of the NetInf architecture in a co-existence scenario introduces new challenges, which are access control and traffic management due to the absence of interaction among the CLs transport functions and NetInf transport functions. Regarding the access control limitation, in NetInf, it is not possible to apply controls over the accessibility levels of the information. Thus, anyone can access the published data without any restriction. The second issue refers to the CLs, which are responsible for the interconnection of different types of networks into a single ICN network. For example, the interaction among the underlying protocols that provide really different communication services (e.g., from uni-directional, opportunistic message forwarding to flow- and congestion-controlled higher layer communication services; from delay-challenged to high-speed optical backbone networks) creates new challenges.

6.3.3 Name Data Networking (NDN) and Content Centric Networking (CCN)

The NDN research project [74] is one of the existing implementations of the CCN paradigm [101], funded by the NSF [5] as part of the FIA program. From its first design late in 2010, the NDN main idea is to shift the existing IP host-to-host communication into a data oriented one by leveraging on an increased responsibility of the routers. Upon receiving a request for a content, the routers first check whether the content is already present in their cache (i.e., Content Store). If this is the case, they immediately return the content back, otherwise, routers check the PIT, searching for a pending request issued for the same content. If the PIT already contains an entry for the specific content, routers just collapse the current request into the PIT. If none of the previous cases verifies, routers forward the request to the next node in the network using the FIB, and keep waiting for the associated data

to return back. Once the data arrives, all the pending interests for that content are satisfied just by sending the copy of data back to all the hosts which have requested it.

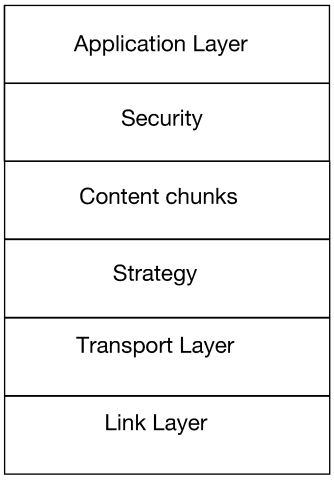

Figure 6.7: Internal architecture of a NDN node in an *overlay* deployment [222] [74].

As shown in Figure 6.7, NDN inherits the hourglass model of the IP architecture, introducing some changes with respect to the original model. In particular, the two novel layers are *Security* and *Strategy*: the first one refers to the NDN design that addresses the security of the content instead of the security of the communication channel between two nodes (which is how IP works); the *Strategy* layer substitutes the network layer and provides the forwarding plane to forward *Content chunks*, giving the best choices to maintain multiple connectivities under varying conditions. In addition, it also supports security, scalability, efficiency and resiliency. Finally, NDN modifies the *Transport Layer* making it consumer-driven instead of producer-driven [42, 43], and it imports it into the NDN forwarding plane.
Deployment Approach: The common implementation of NDN and CCN includes overlay protocols such as CCNx [147] and NDNLP [184], respectively, which are deployed over existing IP infrastructure. For instance, CCNx [161] showcases the explicit example of *overlay* by implementing CCN-over-UDP. In particular, it provides a method to transport CCNx messages between two nodes over UDP. Moreover, a concrete example of NDN overlay architecture is provided by the ndn-testbed[2], which connects multiple NDN

[2]https://named-data.net/ndn-testbed/

nodes located in several continents over existing TCP/IP. The services provided in the trials of CCN/NDN include various project such as real-time video-conferencing [93], adaptive bitrate streaming (not limited to end-to-end) [125, 136, 169] and ndnSIM (NDN simulator module on NS-3) [139].
Deployment scenarios: The *overlay* deployment of NDN over TCP/IP allows the communication between two ICN "islands" through an IP "ocean", as it is confirmed by the ndn-testbed.
Addressed coexistence requirements: NDN guarantees the following three coexistence requirements:

- Forwarding: the router's FIB is responsible for forwarding interests towards the content provider via one or more network interfaces based on the routes to the origin node(s). The requested data packet is then forwarded towards the requester by simply traversing, in reverse, the path of the preceding interest [222]. NDN supports also the multicast data routing, which improves receiver-driven multimedia delivery;

- Storage: NDN routers are enabled to cache contents;

- Security: NDN provides a data-centric security model where each data unit is uniquely signed by the data producer [226].

Architecture or technology used: Besides the standard TCP/UDP/IP tunneling, which is part of the *overlay* approach, the NDN project does not rely on additional architectures.
Evaluation Parameters: The tunneling approach, where NDN/CCN endpoints communicate over IP [20, 146], disowns the fundamental advantages of the content oriented networking (i.e., in-network caching and multicast forwarding). In contrast, the architectures implementing hop-to-hop connection-less (/oriented) connectivity (i.e., over TCP/UDP) suffer from a lack of traffic management [55]. In NDN/CCN networks, Congestion Avoidance (CA) is operated by the consumer rather than by the producer (server) as it is in TCP. It consists of pacing the Interests messages transmission in order to ensure that the delivery of a requested resource can make maximum fair use of the network. Since existing NDN/CCN CA algorithms are largely based on the TCP CA algorithms, they assume that the bandwidth-delay product of the network fluctuates relatively slowly, as all data packets traverse the same path from server to client. However, in NDN/CCN networks content objects may be retrieved from various locations and may reach the consumer through different paths. Thus, the concept of a bandwidth-delay related to a single path and the use of TCP-like CAs do not fits well for

NDN/CCN networks. This is an active area of research in the NDN/CCN community [180].

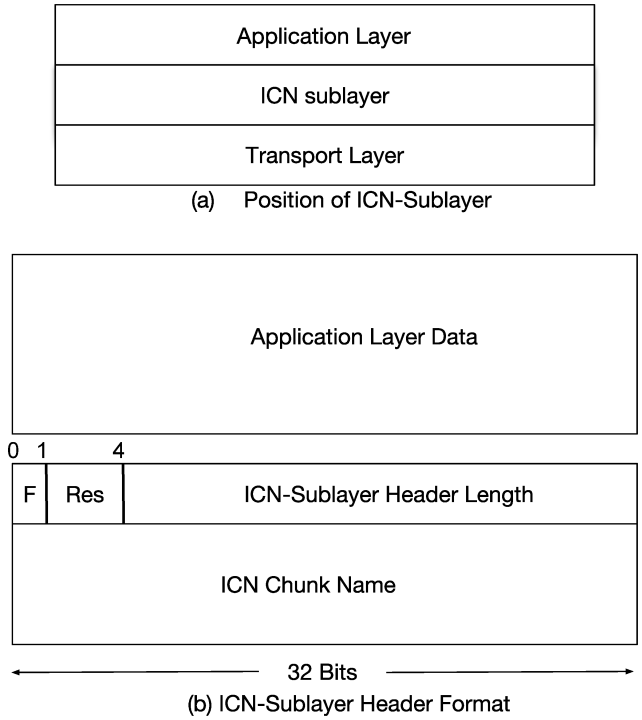

Figure 6.8: Internal architecture of an O-ICN node in an *overlay* deployment.

6.3.4 Overlay architecture for Information Centric Networking (O-ICN)

O-ICN [22, 183] is a novel architecture, which leverages the SDN technology for separating data plane activities (i.e., forwarding and storing/caching of ICN contents) from control plane ones (i.e., naming, name resolution and routing). In particular, O-ICN introduces the ICN Manager as an extended version of a DNS server, which performs name resolution for both ICN and non-ICN requests. In case of an ICN request, the ICN manager identifies the source of the content and sends to it the user's address, so that the source can route back the requested content to the user. In case of a non-ICN request, the standard routing mechanism of TCP/IP is followed. The naming scheme adopted by O-ICN is hybrid, i.e., both human readable and self-certifying as in the SAIL. Finally, the existing routers are modified to cache contents and communicate with the ICN manager.

Figure 6.8 (a) depicts the position of the novel *ICN-sublayer* proposed by O-ICN, which lies between the TCP/IP *Application Layer* and *Transport Layer*. More specifically, Figure 6.8 (b) describes the fields used by the new layer: the ICN flag bit (F), equal to 0 for an ICN request or to 1 for an ICN content; three subsequent bits (1-4) are reserved for additional purposes, and next 28 bits represent the total ICN header.
Deployment Approach: O-ICN relies on an *overlay* deployment solution by leveraging on the ICN Manager, which performs dual tasks: name resolution, along with routing functionalities for ICN requests, and standard DNS resolution for the existing Internet requests.
Deployment scenarios: O-ICN allows the communication between two ICN "islands" through an IP "ocean". Moreover, thanks to the ICN manager capability of manipulating both ICN and not-ICN requests, O-ICN can support also the *Border Island* deployment scenario.
Addressed coexistence requirements: The coexistence requirements addressed by O-ICN are the following ones:

- Forwarding: the ICN Manager is responsible for the forwarding strategy;
- Storage: the data plane activities involve tactical storing/caching of ICN contents at different locations/routers/gateways and are managed by ICN routers.

Architecture or technology used: O-ICN exploits the SAIL solution for the naming scheme and the SDN technology for a separate management of data plane and control plane activities.
Evaluation Parameters: As for the other previous *overlay* approaches, O-ICN is affected from a lack of traffic management. In addition, the ICN manager is not able to guarantee its DNS functionalities in case of dynamic network conditions and the overall solutions suffer from scalability problems.

6.3.5 CONET

CONET [65] is an architecture designed for connecting several CONET Sub Systems (CSSs), which could be the whole Internet network, an IP autonomous system or a couple of network connected components. The main components of the CONET design are the following ones: End-Nodes (ENs), Serving-Nodes (SNs), Border-Nodes (BNs), Internal-Nodes (INs), and Name-System-Nodes (NSNs). An EN requests some named-data by issuing an interest routed by the BNs, which are located at the border of CSSs. The route-by-name process identifies the CSS address of the next

BN, which is closest to the SN as soon as the appropriate CSS is reached. Then, the INs forward the packet using the under-CONET routing engine. The CSS address of EN and the ones of the traversed nodes are appended to the packet, and as soon as a CONET node is found to be able to provide the requested named-data, this is sent back on the reverse path to serve the requesting EN. All BNs and INs along the traversed path may cache the content.

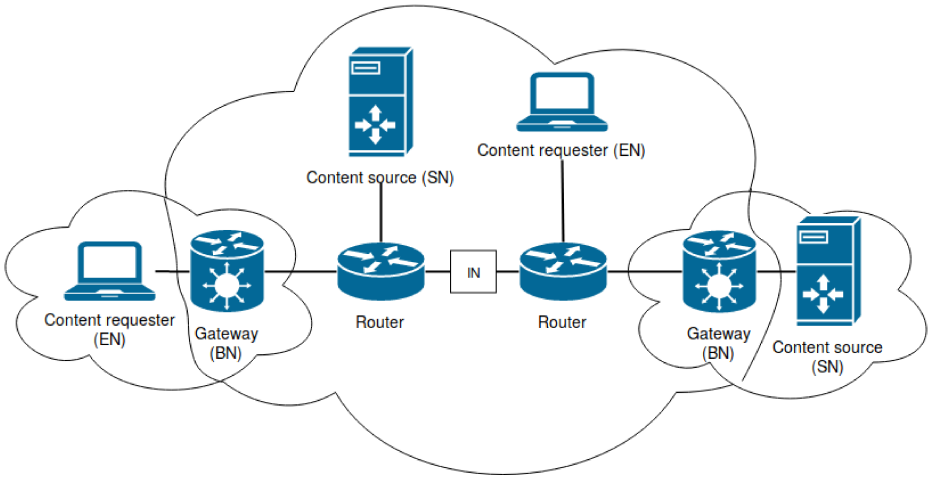

Figure 6.9: Simplified view of the CONET architecture.

Deployment Approach: The CONET architecture can follow either an *overlay* or a *hybrid* deployment approach. In the first case, CONET works on top of the IP layer and the CSSs are nodes connected by overlay links (e.g, UDP/IP tunnels). In the second approach, the purpose is to make IP content-aware by introducing a novel IPv4 option or an IPv6 extension header. The network components will have then hybrid routing tables with both IP network addresses and names.
Deployment scenarios: Considering the *overlay* solution, CONET allows the communication between two ICN "islands" through an IP "ocean". On the contrary, the *hybrid* approach allows it to be deployed in the *Border Island* scenario as well.
Addressed coexistence requirements: CONET guarantees the following three coexistence requirements.

- Forwarding and Management: forwarding is guaranteed by BNs and NSNs. In addition, ENs provide transport-level functionalities such as

reliability and flow control. Since the logic for requesting a content involves sending separate interests, each one containing a small part of the named-data, the control of interest sending rate can be used as a TCP-like flow control mechanism;

- Storage: BNs are able to store contents.

Architecture or technology used: Besides the standard TCP/UDP/IP tunneling, which is part of the *overlay* approach, the CONET project does not rely on additional architectures.
Evaluation Parameters: The *hybrid* deployment solution is hard to be introduced since it requires a new IP option. However, with respect to the *clean-slate* approach, the *hybrid* one is less disruptive, and it allows the architecture deployment in different scenarios.

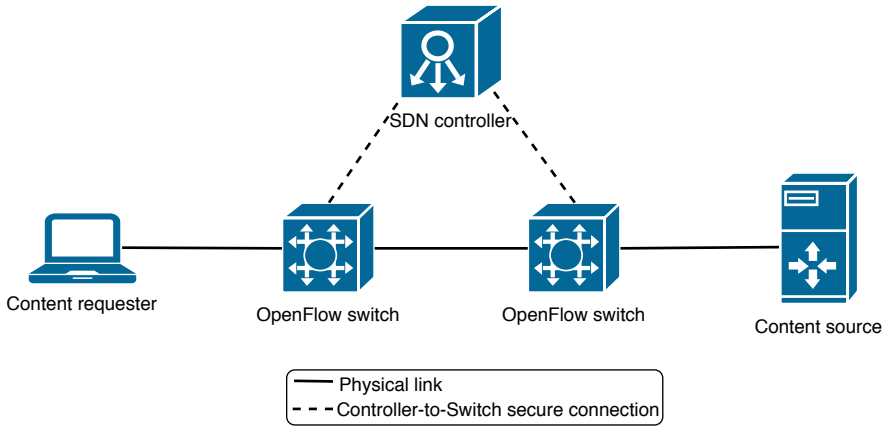

Figure 6.10: Simplified view of the solution proposed by Vahlenkamp et al. [198].

6.3.6 GreenICN

The SDN technology decouples control plane from data plane, and it provides a programmable, centrally managed network control that improves network performance and monitoring. SDN-based implementations of ICN exploits the centralized view available to SDN controller, which enables the SDN controller to install appropriate forwarding rules for ICN requests/responses in such a manner that the network elements only have to support IP forwarding. Vahlenkamp et al. in [198] proposed an implementation of ICN using SDN under their GreenICN project. The proposal leverages ICN protocol's Message IDs and features of SDN instantiations such as OpenFlow to rewrite packet header information. Figure 6.10 presents a

simplified view of this solution. Here, both the *content requester* and the *content source* are connected to *OpenFlow-enabled switches* that are managed by the *SDN controller*. Routing information for the content requests and responses upon arriving on OpenFlow switches is handled/rewritten by the instructions from the controller.
Deployment Approach: The proposed solution is an *overlay* ICN implementation ICN messages are transferred with existing UDP or TCP.
Deployment scenarios: Essentially, the authors in [198] propose ICN deployment over IP network, where an ICN-aware content source delivers the content to an ICN-aware *requester* over IP network. Hence, this solution enables the communication between two ICN "islands" through an IP "ocean".
Addressed coexistence requirements: The architecture addresses the following coexistence requirements:

- Forwarding: network programmability offered by SDN enables forwarding and routing for ICN;
- Management: SDN centrally managed network control supports load-balancing, traffic engineering, and explicit path steering.

Architecture or technology used: The authors argue that an ideal or native deployment of ICN, in which user devices, content sources, and intermediary network elements are ICN aware, may not be viable. Hence, the authors proposed to implement ICN-awareness in the SDN-enabled switches, where ICN packets are carried over the IP transport protocol. By using SDN, the authors target all the services/applications of the TCP/IP protocol stack.
Evaluation Parameters: In the proposed ICN implementation, SDN controller must manage every ICN request and rewrite several headers fields for every response packet, which might not scale with increased network size. Given that this solution is based on the widely accepted SDN technology - that supports agile deployment and rapid alternation in networking - the hardware modifications required for its deployment are low in those scenarios where SDN infrastructure already exists. Consequently, the time required for its deployment are also low. Nevertheless, the time and the hardware modifications required for its deployment would be higher if the SDN infrastructure does not already exists.

6.3.7 coCONET

Similar to the work in [198], Veltri et al. [202] proposed a CONET [65] inspired by SDN-based implementation of ICN, called coCONET. Figure 6.11 presents a simplified view of this solution. In this architecture, ICN nodes

and user-terminals form the data plane and Name Resolution Service (NRS) nodes are placed in the control plane. Moreover, ICN node works as an OpenFlow switch while NRS node works as an OpenFlow controller.

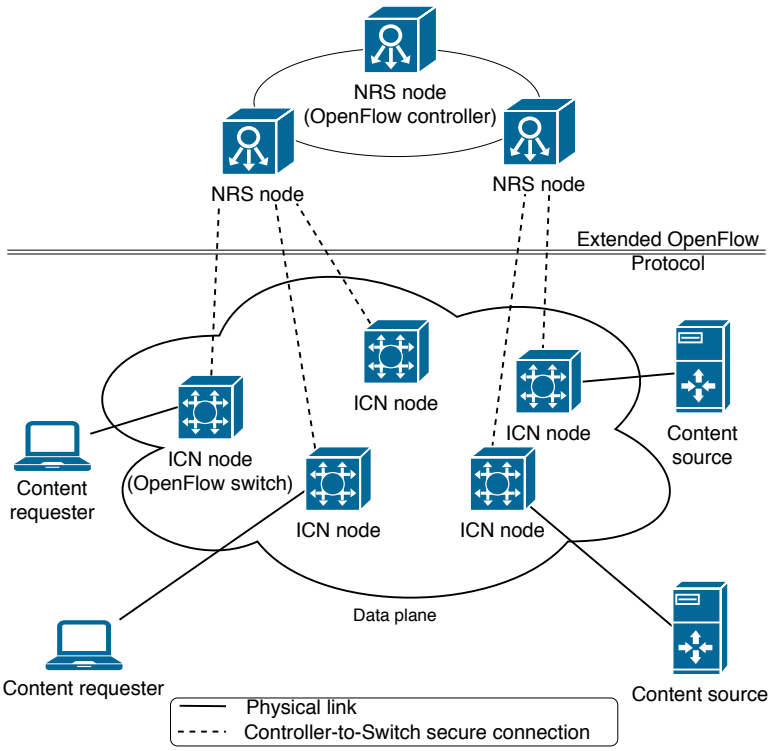

Figure 6.11: Simplified view of the solution proposed by Veltri et al. [202].

Deployment Approach: Similar to the work [198], the proposed solution is an *overlay* ICN implementation.
Deployment scenarios: The proposed solution enables the communication between two ICN "islands" through an IP "ocean" and between an ICN "island" and an IP "island" through an IP "ocean", where the underlying IP network is managed by OpenFlow-based SDN network.
Addressed coexistence requirements: The present architecture provides the following coexistence requirements:

- Forwarding and Management: SDN-based operations of the proposed approach support both forwarding and management of ICN traffic;
- Storage: ICN capable nodes cache the contents;
- Security: Contents are cryptographically protected in order to assure content (and content generator) authentication and data integrity.

This security service is provided through digital signature and can be verified through the public key associated to the private key of the content (or of the content generator). The proposed system enforces every ICN node to verify such signature before forwarding the content toward the interested end-nodes, to protect the network against DoS or other attacks.

Architecture or technology used: Here, the authors focus specifically on OpenFlow [141] based SDN implementations and target all the services/applications of the TCP/IP protocol stack. OpenFlow is a flavor of SDN.
Evaluation Parameters: The proposed solution requires ICN capable OpenFlow network devices for ICN operations. Due to such specific requirements, the hardware modifications and the time required for its deployment are high.

6.3.8 DOCTOR

DeplOyment and seCurisaTion of new functiOnalities in virtualized networking enviRonments (DOCTOR) [2] is an ongoing project funded by French Nation Research Agency (ANR). The project provides support towards the adoption of new standards by developing a secure use of virtualized network equipment. This leads to ease the deployment of novel networking architectures, thus enabling the coexistence of IP and emerging stacks such as NDN as well as the progressive migration of traffic from one stack to the other. DOCTOR proposes the use of NFV infrastructure to achieve the incremental deployment of NDN at low cost. The project proposes a HTTP/NDN gateway to interconnect ICN "islands" to the IP world, and an experimental architecture supporting this research direction and able to process the web traffic passing through a virtualized NDN network.

In particular, DOCTOR first deploys a virtual network based on OpenvSwitch to provide an end-to-end network connectivity between the virtualized network services and to enable a software control of the networking infrastructure. Then it selects NDN as an ICN protocol stack, specifically, the NDNx software is *dockerized* to make it a Virtualized Network Function (VNF), deployable in DOCTOR architecture. In DOCTOR, NDN is used both over IP and over Ethernet. It is because the most of NFV tools are still IP-dependent. To test the functionality of the coexistence, the web is considered as application-layer service due to its high-popularity and predominance in the global network shares. However, since current web clients and servers do not yet implement NDN, dedicated gateways are being used

to perform an HTTP/NDN conversion. These gateways are conceived as VNFs, thus it can be deployable where and when required. In particular, two types of gateways are defined: (1) an ingress gateway (iGW), aiming at converting HTTP users requests into NDN Interest messages to find the content in the NDN network, and converting NDN Data messages into HTTP replies sent to the end-users and (2) an egress gateway (eGW), aiming at converting NDN messages into HTTP requests towards IP websites if the content is not available in the ICN network, and converting HTTP replies into NDN Data messages to the iGW. Figure 6.12 shows the highlevel architecture a virtualized node in DOCTOR. The virtualized node that runs on commodity machines is implemented on a single Linux server, and it provides the hardware resources for virtualization as required for supporting the execution environment for Virtualized Network Functions (VNFs). The VNFs shown in Figure 6.12 can act various components such as as NDN stack, IP stack, and HTTP/NDN gateway, which enables the coexistence of IP and emerging stacks (e.g. NDN) as well as the progressive migration of traffic from one stack to the other.

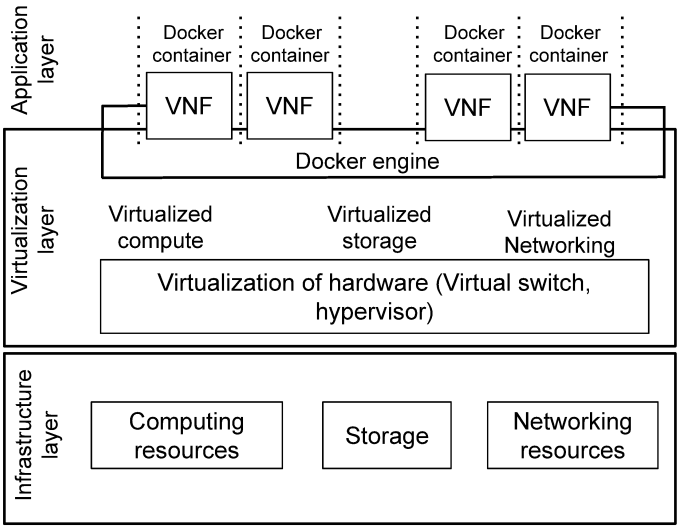

Figure 6.12: Doctor virtualized Node architecture

Deployment Approach: DOCTOR uses an *underlay* approach with the help of an HTTP/NDN gateway, that can map the HTTP protocol with NDN messages and properly deliver the web content.

Deployment scenarios: The ingress and egress gateways in DOCTOR allows it to operate in IP-to-ICN and vice-versa in IP and ICN "islands". In addi-

tion, it also supports *Border Island* deployment scenarios, thus it is able to exchange communication through both " oceans".
Addressed coexistence requirements: The DOCTOR architecture addresses the following coexistence requirements:

- Forwarding: explicit name based routing of NDN is performed at each router through the use of virtualized NDN stack;
- Storage: CS performs the content caching;
- Security: content oriented security of NDN is supported by the architecture;
- Management: For efficient network management, the DOCTOR's Control and Management plane for management and orchestration of VNFs has been designed with respect to the recommendations of the ETSI NFV group concerning the NFV Management and Orchestration (NFV-MANO).

Architecture or technology used: The architecture of DOCTOR is flexible as it is based on NFV and SDN principles. The main component of the architecture is the NFV Infrastructure (NFVI), which enables the resource virtualization and management to host VNFs deploying an ICN protocol stack on the data plane, and the control plane defines the management and orchestration (MANO) aspects. As a computing virtualization framework, the architecture chooses Docker, which relies on a lightweight virtualization principle. Even if Docker offers lower isolation guarantees, this choice was driven by the better performance compared to other virtualization solutions.
Evaluation Parameters: One of the key issue raised in the solution proposed by DOCTOR that provides HTTP-level translations into ICN semantics is *latency*, which occurs due to repeatedly sending requests to the ICN servers which acts as the gateways and are attached to source such as content provider. Here, the problem lies in the fact that URI is likely different for every content. Therefore, for each new published content, it represents a new routing identifier which should be updated to the gateways. This results in a continues interaction of content publisher and gateways for each HTTP-get request to be routed accurately in the network, thus results in an additional overhead and network latency.

6.3.9 iP Over IcN- the betTer IP (POINT)

POINT [6] is a European Project, funded under the H2020 programme, started in January 2015 and ended in December 2017. Its main purpose is

to evaluate both quantitatively and qualitatively the improvements introduced by running ICN over an IP network. To achieve this aim, POINT designs an evolution of the PURSUIT architecture, which both leverages on the SDN technology and on additional network components that enable IP-based applications to run in the new setup without any modification. Those new elements are the network attachment points (NAPs) and the ICN border gateways (ICN BGWs). The former directly interact with the end user devices and are responsible for the translation of all the IP protocol abstraction layers (i.e., HTTP, TCP, COAP and IP) into the ICN paradigm, while the latter control the communication between ICN and IP networks. Furthermore, the NAP provides standard gateway functions such as Network Address Translation (NAT), firewall, and dynamic IP address assignment. The core ICN functionalities are provided by the PURSUIT components (i.e., TM, FN, and RV). Usually, content items are assigned a *Rid* and are stored on the publisher, which advertises the contents availability in the network. Then, a user device sends a request for a content item and the NAP transforms the interest into a subscription for a specific *Rid*. The subscription is then sent to the RV, which triggers the TM towards the identification of a path between publisher and subscriber. The TM identifies all the nodes that need to be traversed and it calculates the associated FIDs, which are placed in the packet header. At this point, the SDN switches are responsible for forwarding the packets by using only the FIDs and not the routing tables. The SDN switches are not aware of the POINT architecture and are, instead, coordinated by an SDN controller, which communicates directly with the TM. This communication is bidirectional since the SDN controller informs the TM about any topology modifications, and the TM notifies the SDN controller about the configuration to be placed on the SDN switches.

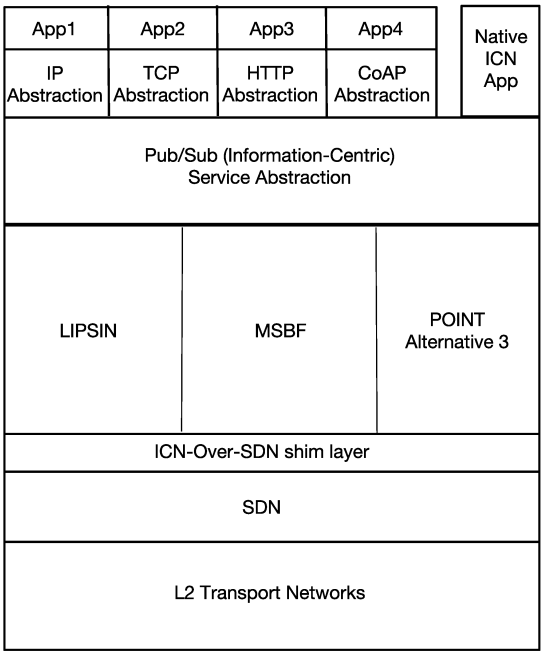

Figure 6.13: Internal architecture of a POINT node in an *underlay* deployment

Figure 6.13 shows a simplified version of a POINT node internal architecture. In the upper layer of the node, there are generic applications (i.e., *App1*, *App2*, *App3*, *App4*) which interact with a set of abstractions provided by the POINT architecture (i.e., *IP Abstraction*, *TCP Abstraction*, *HTTP Abstraction*, *CoAP Abstraction*). Those are aimed at enabling the communication between applications and ICN networks without requiring any modification from the application interface side. Each abstraction, then, cooperates with the *Pub/Sub (Information-centric) Service Abstraction* to adhere to a publish/subscribe paradigm, where information is delivered according to specific strategies (i.e., *LIPSIN*, *MSBF*, *POINT Alternative3*). Finally, POINT exploits also the SDN technology by introducing two new layers (i.e., *ICN-over-SDN shim layer* and *SDN*) just above the *L2 Transport Network* layer.

Deployment Approach: The POINT project falls under the *underlay* deployment approach due to the gateway components, which are responsible for the translation from the IP semantics into the ICN.

Deployment scenarios: The main purpose of the POINT architecture is to enable different subnetworks to communicate among each other. Thus,

POINT enables the communication between an ICN "island" and an IP "island" through an IP "ocean".
Addressed coexistence requirements: Given that POINT is an evolution of PURSUIT, they both share the same coexistence requirements, i.e., forwarding, storage, and security.
Architecture or technology used: The POINT solution relies on both the PURSUIT architecture and the SDN technology.
Evaluation Parameters: The challenges introduced by the POINT project involve management of dynamic networks, scalability and latency of data transmission. The first two challenges refer to the appropriate configuration of SDN switches to face an automatic update of network topology (e.g., a new host being attached). On the contrary, the third challenge might be due to the high frequency of interaction between NAP and RVs.

6.3.10 architectuRe for an Internet For Everybody (RIFE)

The RIFE [1] architecture is a Horizon2020 funded project, which started in February 2015 and ended in January 2018. Its aim is to develop a new network infrastructure that brings connectivity to communities living in remote locations or unable to afford the communication network costs. To achieve the purpose, the RIFE project focuses on three different challenges regarding the current end-to-end communication paradigm: reduction of capacity, energy, and redundant contents available in the network. The first one can be achieved through a time-shifted access to network services and applications, it also enables disconnected environments to have access besides an additional delay. The energy consumed by connected devices can be reduced by introducing a tolerance delay in the communication, so that devices can stay in an idle mode during the absence of network activity. Finally, the third aim is achievable by serving the same content to all the clients that requires it, instead of releasing each time a new copy. The architecture addressing those objectives is a combination of IP, ICN, and DTN paradigms.
Deployment Approach: The RIFE architecture follows the *underlay* approach because of the gateway components, which are responsible for the translation from the IP semantics into the ICN one.
Deployment scenarios: RIFE enables the communication between an ICN "island" and an IP "island" through an IP "ocean".
Addressed coexistence requirements: RIFE is an evolution of the PURSUIT architecture. Thus, the coexistence requirements addressed are the same, i.e. forwarding, storage, and security.

Architecture or technology used: The architecture proposed in the RIFE project is a modification of the PURSUIT one and it relies on the coexistence of IP, ICN and DTN. This last architecture is responsible for introducing the delay and disruption tolerance required to enable the time-shift requirement.
Evaluation Parameters: No additional challenges have been found for the RIFE project, apart from above-mentioned *underlay* limitations.

6.3.11 CableLabs

Among the different *underlay* approaches, there is a solution designed by CableLabs, which is a non-profit Innovation and R&D lab focused on the introduction of fast and secure release of data, video, voice, and services to end users. CableLas proposed an incremental introduction of CCN/NDN in the existing CDNs to improve the overall content distribution without modifying IP routers [207]. The architecture designed by CableLabs requires first a migration of some services/applications to the ICN paradigm, and then the introduction of proxies. Those are able to manage the translation between HTTP and CCN. Once several ICN "islands" are deployed in the network, the communication among them is provided through IP tunneling.
Deployment Approach: The solution proposed by CableLabs adopts the *underlay* approach because of the gateway components, which are responsible for the translation from the IP semantics into the ICN one.
Deployment scenarios: The CableLabs architecture supports the communication between an ICN "island" and an IP "island" through an IP "ocean".
Addressed coexistence requirements: The CableLabs architecture addresses the following coexistence requirements:

- Forwarding: the additional proxies introduced in the network to support the translations i.e., HTTP to CCN and CCN to HTTP, also work as CCN forwarder;
- Storage: as the architecture is an evolution of a CDN, by design the network nodes can cache contents.

Architecture or technology used: Throughout this project, CableLabs investigated how the CCN infrastructure is better in supporting a content-oriented network with respect to the current solutions, such as CDNs. Thus, CableLabs illustrates an incremental deployment of a CCN network over existing CDN.
Evaluation Parameters: The following ones are the challenges identified by CableLabs with respect to their own architecture: optimized CCN router implementation (e.g., FIB/PIT sizing and memory bandwidth), optimized

CCN cache implementation, congestion avoidance (i.e., a bandwidth-delay approach for requesting CCN resources is not applicable given the different locations and paths that can be traversed), cache control and semantics (i.e., identification of an appropriate protocol for deleting objects in a cache), content object size and fragmentation (i.e., definition of the maximum content object size transmissible inside a network), network control (e.g., update cache location which means updating routing policies), CCN to HTTP and HTTP to CCN conversions (e.g, the computational complexity of the translation function), business rules and monetization (e.g., criteria for monetizing the distribution of proprietary content).

6.3.12 NDN-LAN

The authors in [211] propose a *hybrid* ICN architecture in which content names are mapped to the MAC addresses. In particular, the authors present the design of a Dual-Stack switch (D-switch), which provides name-based forwarding for NDN traffic and address-based forwarding for conventional traffic such as IP. It can be seen from Figure 6.14 that the key component of D-switch architecture is *dispatcher*, which checks the EtherType field in the header of a received frame. When an IP frame is detected, the D-switch works like a traditional Ethernet switch and it forwards the frame using the MAC address. If an NDN frame (i.e., interest or data packet) is detected, the D-switch processes/forwards the frame based on the content name carried in the NDN header (i.e., layer 3). In particular, the dispatcher either selects the *process IP traffic* or *process NDN traffic* module in the D-switch based on the value of EtherType field. In *process NDN traffic* module, the PIT and FIB tables are modified to store the mapping between the content names and MAC addresses. For instance, when an Interest packet is received, the D-switch will forward it by searching the content name and its corresponding MAC in the FIB, and then fill the destination MAC address field in Ethernet header with the recorded MAC address.

Deployment Approach: This coexistence approach falls under the *hybrid* Deployment Approach because the D-switches are able to process both types of traffic (i.e., IP and NDN). In particular, a LAN consists (fully or partially) of D-switches that can process the data traffic received from NDN-enabled hosts as well as IP hosts. However, a fully hybrid scenario needs to be consistent with D-switches only, else other techniques or polices/rules are required to perform the data forwarding.

Deployment scenarios: The D-switches allow NDN traffic to run within the IP network, so it enables the communication between two ICN "islands"

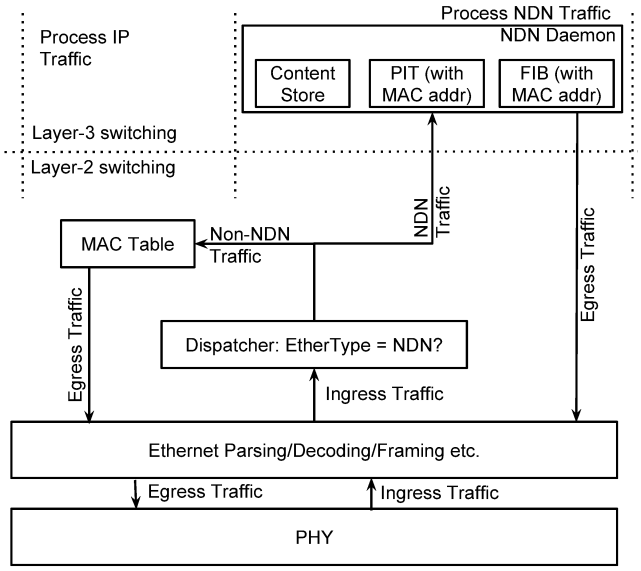

Figure 6.14: Dual-stack switch architecture

through an IP "ocean". However, due to the use of MAC-layer encapsulation only, the inter-network communications are not possible, so it does not provide support for the *Border island* scenario.
Addressed coexistence requirements: The present architecture provides the following coexistence requirements:

- Forwarding: Full advantage of ICN features such as in-network caching and native multicast is supported when the underlying LAN consists of D-switches only. However, when the LAN has both D-switch and E-switch, it has to be carefully designed to avoid conflict between name-based forwarding and address-based forwarding.
- Storage: In-network caching is only supported at D-switches, and it is the responsibility of the network manager that the E-switches do not receive an ICN packet because such packet will be discarded.
- Management: Management of such a deployment is challenging due to limitations of topology creation and forwarding rules installation.

Architecture or technology used: NDN-LAN is mainly suitable for NDN applications that run in small and private networks such as university campus and within an organization. However, the proposed coexistence solution aims to support a variety of applications which includes NDN as well as IP applications. It is done by achieving the following design goals: (i) coexistence with IP traffic, it ensures that the common mechanisms should run

without any change or performance penalty, (ii) native NDN support, by not relying on tunnels or overlays, and (iii) incremental deployment and general applicability. The proposed solution does not make use of any specific technology to implement the Dual-switch logic. Minor hardware and software changes in the Dual-switches allow them to process the IP and NDN traffic in a controlled environment (i.e., LAN).

Evaluation Parameters: To implement the required logic and functionalities at D-switches so that it can support NDN-enabled traffic processing, some changes are required in the switch hardware as well as software. Additional forwarding polices need to be installed in scenarios where D-switches coexist with the E-switches. Without any standardization of these new software/hardware components, the applicability of the proposed solution in real-world coexistence applications is limited. The design of mechanisms to support name-based forwarding, meanwhile coexisting with address-based forwarding within the same LAN, is a challenging task. Additionally, the process for D-switches to learn the forwarding table at layer-2 and build name-based FIB at layer-3 is an open problem that needs to be addressed. In LANs, the implementation of the proposed solution is simple and straightforward, however, as the LAN size increases and communication between different LANs are needed, the deployment cost will increase significantly, and the current solution needs to be extended to deal with new issues such as interoperability and scalability.

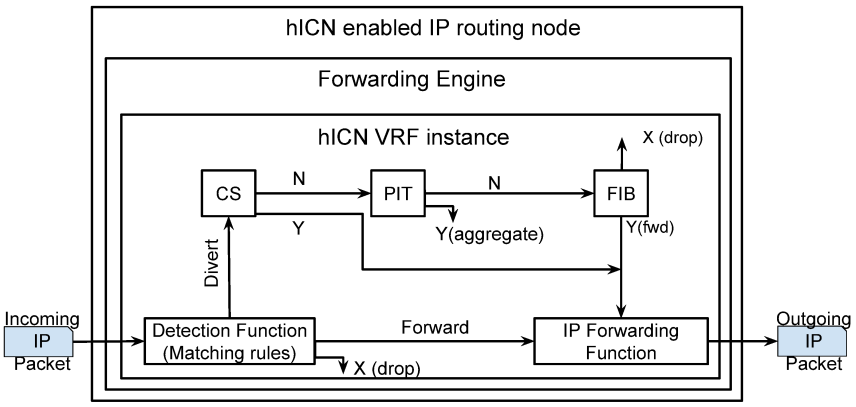

Figure 6.15: hICN node architecture

6.3.13 Hybrid ICN (hICN)

Authors in [150] propose methods and systems to facilitate the integration of ICN into IP networks. The hICN communication system claims to have

the ability to preserve ICN features and advantages, while, at the same time, benefiting from exploiting an existing IP infrastructure. The major components of hICN communication system are as follows: (i) *hICN-enabled IP router(s)*, capable of processing and forwarding both regular IP packets and IP packets enhanced with ICN semantics, (ii) *IP router(s)*, capable of handling IP packets, and (iii) *hICN router(s)*, being provisioned with a consumer or producer application. The traditional IP packet headers have been modified to add the ICN semantics. As it is seen in Figure 6.15 that when a router receives an IP packet, then based on the IP header content, it can identify how to process it, i.e., using ICN or IP stack. The authors suggest two possible name mapping schemes for hICN content names to IP: (i) *Pure IP mapping*, in which content name components can be directly encoded in the IP header, and (ii) *Optimized mapping*, in which a subset of the content name component is encoded at network header while the remainder is encoded in the transport header.

Deployment Approach: As the hICN-enabled IP routers are capable in processing IP as well as ICN traffic, this approach falls into the *hybrid* deployment Approach. However, unlike NDN-LAN, in which MAC-to-content name mapping and conversely is performed, in hICN, the IP-to-content name and conversely is done.

Deployment scenarios: Due to the presence of dual stack routers, the proposed architecture supports all the deployment scenarios.

Addressed coexistence requirements: hICN can be considered as one of best proposals to support the coexistence because it retains most of the ICN basic features (e.g., layer-3 name-based routing, partial symmetric routing, object-based security, anchorless mobility, and in-network reactive caching). This is because hICN exploits the IPv4 and IPv6 header fields content semantic to identify whether the received packet is an IP data packet or an IP interest packet. The use of IPv4 or IPv6 RFC compliant packet formats guarantees the communication between an IPv4/IPv6 router and a hICN one. More specifically, the hICN router processes and forwards both the regular IP packets and the ICN-semantic-based ones. Hence, it preserves pure ICN behavior at layer-3 and above by guaranteeing end-to-end service delivery between data producers and data consumers using ICN communication principles. The present architecture provides the following coexistence requirements:

- Forwarding: The hICN-enabled IP routers as well as IP routers uses the same forwarding module.

- Storage: The cache stores are available on hICN-enabled IP routers, and the interest packets could be satisfied by these routers if the requested content is available in the router cache.
- Management: For large scale usage of this architecture, the consumer and producer applications must have the mapping of content-names with the corresponding IP addresses, so that the ICN packets can be processed seamlessly by the non-ICN enabled routers as well.
- Security: The architecture provides the same security features that are provided by ICN. However, the IP-only routers are not able to check the received data packets integrity and authentication, hence, atleast one hICN-enabled IP router must be available at the route between the consumer and producer.

Architecture or technology used: The hICN proposal uses the IP packet header semantics to differentiate the ICN and IP packets, and the mapping table at hICN-enabled router or DNS is used for performing the mapping task. To support interoperablity between different networks, the edge route could translate the incoming packet to hICN compliant packets using a proxy. Therefore, hICN does not use any specific architecture (e.g., SDN) or technology (e.g., virtualization or tunnelling) to perform the coexistence. *Evaluation Parameters:* The major challenge of hICN are similar to the other hybrid approaches and it includes a lack of support for heterogeneity, scalability, and standardization of the proposed changes in traditional Internet protocols and network components. Additionally, the communication delay caused by the additional time taken at hICN routers for the mapping could be an issue for delay sensitive applications. The hardware modifications are minimal because the hICN routers can be created by installed a software bundle in the existing IP routers, however the memory requirements will increase due to the need of storage cache. The deployment effort will be considerable due to the need of the modifications in the consumers and producers applications.

6.3.14 Melazzi et al. [143]

The authors proposed an SDN-based *hybrid* implementation of ICN. The proposed approach is an extension of the CONET architecture [65] for OpenFlow networks, where dedicated border nodes perform name-to-location resolution, using an external system, for any requested Named Data Object (NDO). Figure 6.16 presents a simplified view of this solution. The authors propose to include two different forwarding strategies in an ICN node: (1)

to forward content requests; and (2) to deliver the data. *Forward-by-name* feature of an ICN node applies to Interest packets while *Data Forwarding* is the mechanism that allows the content to be sent back to the device that issued a content request. *Content Routing* used to disseminate information about location of contents, and *Caching* is the ability of ICN nodes to cache data and to directly reply to incoming content requests.

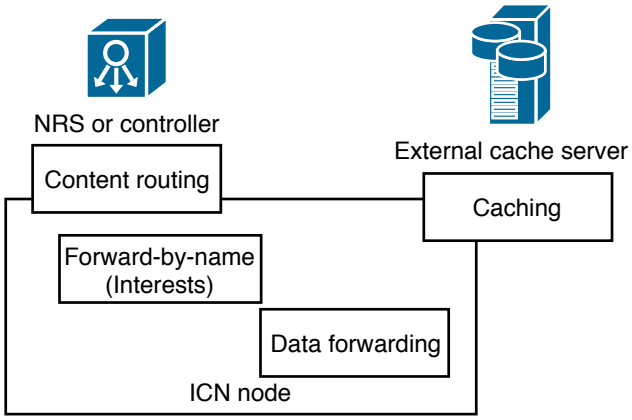

Figure 6.16: Simplified view of the solution proposed by Melazzi et al. [143].

Deployment Approach: The proposed approach is a *hybrid* approach.
Deployment scenarios: The proposed implementation of ICN is an extension of the CONET framework, in which *border* nodes interconnect different CSSs. Hence, this solution can be categorized as *Border Island*-based ICN deployment.
Addressed coexistence requirements: The proposed system is based on CONET framework. Extending the primary goals of CONET framework, this architecture aims to support forwarding, storage, management, security, and interoperability for ICN deployment.
Architecture or technology used: The present solution strongly relies on the architecture proposed in the CONET project and, through SDN/OpenFlow, it targets all the services/applications of the TCP/IP protocol stack.
Evaluation Parameters: The architecture of the solution requires the networking elements to be OpenFlow complaint. Given that OpenFlow (SDN) has been widely adopted in the networking domain, the hardware modifications and the time required for its deployment are low in scenarios where OpenFlow-based network is already present. On another side, the hardware modifications and the time required for its deployment would be more if OpenFlow-based network is not already present.

6.4 Lessons Learned and Research Directions

The purpose of this section is to summarize the findings achieved through the analysis and comparison of all the existing coexistence architectures (Section 6.4.1) and to finally conclude with some future research directions concerning the coexistence between the current and the future Internet architectures (Section 6.4.2).

6.4.1 Lessons Learned

The main aim of this survey is to provide the necessary overview of the available solutions that already address the coexistence to move the research community towards the design of the most appropriate architecture that enables the transition towards the future. Thus, to guide the reader towards the interpretation of the Table 6.1 which we described in Section 6.3, we add here two new tables, which are a summary of the previous one. In particular, we thought that among all the features and evaluation parameters considered in this survey, the only ones that can be chosen by a network designer are the *deployment approach* and the possible *architecture or technology* to be used in the design of his solution. Thus, Table 6.2 and Table 6.3 are aimed at comparing each *deployment approach* and each *architecture or technology used* with respect to all the other features and evaluation parameters, respectively. As a matter of fact, the *deployment scenarios*, as well as the *addressed coexistence requirements*, directly depend on the *deployment approach* or on the *architecture or technology*, while the evaluation parameters are dynamic properties which might be evaluated during the runtime deployment of an architecture.

The content of the cells, as well as its meaning, is shared between Table 6.2 and Table 6.3. More specifically, the content of each cell corresponds to the number of coexistence architectures addressing both the properties specified in the corresponding row and column (e.g., in the first cell of Table 6.2 the value equal to 7 means that there are 7 coexistence architectures adhering to the *overlay* approach and supporting the *forwarding* functionality). The meaning of the values in the cells is different throughout the table. In the upper part (i.e., rows referring to deployment scenarios and addressed coexistence requirements), the higher is the number, the more positive it is. On the contrary, in the lower part of the table (i.e., rows referring to the evaluation parameters), the higher is the number, the worse it is since the rows refer to the limitations of an architecture.

Table 6.2: Comparison of all the deployment approaches for coexistence architectures.

		Deployment Approach		
		Overlay	Underlay	Hybrid
Addressed coexistence requirements	Forwarding	7	4	4
	Storage	6	4	4
	Security	4	3	2
	Management	3	0	3
Deployment scenarios	ICN-ICN communication in IP "ocean"	7	1	3
	ICN-IP communication in IP "ocean"	2	2	2
	ICN-IP communication in ICN "ocean"	0	2	2
	IP-IP communication in ICN "ocean"	0	2	2
	Border Island	2	3	3
Evaluation parameter	Traffic management	4	1	1
	Access control	1	1	0
	Scalability	2	2	2
	Dynamic network management	1	2	1
	Latency	0	2	2
	Other	4	6	2

The value of each cell refers to the number of coexistence architectures addressing both the properties specified in the corresponding row and column.

Table 6.3: Comparison of all the architectures and technologies used in coexistence architectures.

		Architecture or technology used								
		PSIRP	LAN	SAIL	SDN	PURSUIT	CDN	DTN	CONET	DNS
Addressed coexistence requirements	Forwarding	1	2	1	6	2	1	1	1	1
	Storage	1	2	1	5	2	1	1	1	1
	Security	1	1	0	4	2	0	1	1	1
	Management	0		0	3	0	0	0	1	1
Deployment scenarios	ICN-ICN communication in IP "ocean"	1	2	1	4	0	0	0	0	1
	ICN-IP communication in IP "ocean"	0	1	0	3	0	1	0	0	1
	ICN-IP communication in ICN "ocean"	0	1	0	1	0	1	0	0	1
	IP-IP communication in ICN "ocean"	0	1	0	1	0	1	0	0	1
	Border Island	0	0	1	4	2	0	1	1	1
Evaluation parameter	Traffic management	1	2	1	1	0	1	0	0	0
	Access control	0	0	0	1	0	0	0	0	0
	Scalability	0	1	1	3	2	0	1	0	1
	Dynamic network management	0	1	1	2	2	0	1	0	0
	Latency	0	1	0	1	2	0	1	0	1
	Other	0	0	0	3	0	6	0	1	0

The value of each cell refers to the number of coexistence architectures addressing both the properties specified in the corresponding row and column.

Table 6.2 shows on the columns the three different deployment approaches (i.e., *overlay*, *underlay* and *hybrid*), while on the rows there are all the other features, except for the architectures or technologies used, considered in Table 6.3. Considering the deployment approaches, we found seven architectures adhering to *overlay*, four to *underlay* and the remaining four *hybrid*. As it is shown in the table, a plausible reason for this greater adoption of the *overlay* approach with respect to the other ones might be the higher number of addressed coexistence requirements provided by it. As a matter of fact, almost all the *overlay* architectures guarantee the forwarding and storage features and the number of the ones supporting security and management is higher than in the *underlay* and *hybrid* cases. On the contrary, adopting an *overlay* approach prevents architectures from being deployed in all the *deployment scenarios*: none of the *overlay* architectures covers either the *ICN-IP communication in ICN "ocean"* or the *ICN-IP communication in IP "ocean"* scenarios. Finally, considering the *evaluation parameters*, the most of *overlay* architectures are not able to properly manage the network traffic, but the other limitations are comparable with the ones affecting the *underlay* and *hybrid* solutions. Moreover, even if the number of challenges under the last class (i.e., *Other*) might be significant, we remind that those limitations strongly depend from the design of each coexistence architecture.

6.4.2 Research Directions

As confirmed by the large number of coexistence projects (e.g., POINT, DOCTOR, and hICN) that we surveyed in this chapter, the Industry and Government are pushing towards the definition of a new Internet architecture (i.e., ICN) and its coexistence with the current architecture (i.e., IP). Over the years, the research community has significantly grown around ICN, following different coexistence design approaches. Since a clean slate deployment of ICN requires overhauling the entire Internet infrastructure and changing all the host and producer applications, it is difficult to simply transit from research testbeds to operational networks. Based on the experience received from the initial ICN architecture efforts (e.g., NDN), researchers have realized that it is difficult, as well as infeasible, to replace a greatly successful imperative architecture with a clean slate approach. A plausible reason for this is that ICN remains unproven due to the lack of large scale testbeds, and the consequently limited number of users in a trial, and that it has been tested on a limited number of applications so far.

In the literature, three main approaches (i.e., *underlay*, *overlay*, and *hybrid*) have been used to deploy coexistence architectures. The *underlay* approach introduces communication latency due to the required mapping between IP and name addresses, which limits its usability for real-time and delay-sensitive applications. On the contrary, the *underlay* approach maintains an unaltered quality of service under both normal and exceptional conditions, such as failure, server and link congestion, which are common in operator networks. Considering the *overlay* approach, a major drawback is that it requires the definition and standardization of a new packet format, together with protocols that manage the mapping between ICN faces and IP addresses in the ICN routers FIB. Thus, *overlay* poses a significant challenge to network operators and developers. Additionally, upon new deployment, the tunnel configurations in *overlay* needs to be manually changed to include the newly deployed ICN nodes, and these point-to-point tunnels limit the ICN capability in utilizing the underlying broadcast media. Finally, the *hybrid* approach offers an interesting alternative as it allows ICN semantics to be embedded in standard IPv4 and IPv6 packets so that the packets can be routed through either IP routers or hybrid ICN routers. However, the detailed performance results for *hybrid* solutions are still incomplete, which limits its usage in real deployment scenarios.

In the past few years, a significant effort put by Governments, Industry and Academia to assess the feasibility and effectiveness of ICN indicates that ICN paradigm is being considered as a possible replacement for the current IP-based host-centric Internet infrastructure. Hence, we now present few research directions that could be exploited in this direction.

- *Secure transition phase:* From its start, ICN was purposefully designed to have certain inherent security properties such as authentication of delivered content and (optional) encryption of the content. Additionally, relevant advances in the ICN research community have occurred, promising to address each of the identified security gaps. However, due to the lack of real deployments, an array of security features in ICN networks are still under-investigated, including access control, security of in-network caches, protection against various network attacks (e.g., DDoS), and consumer privacy. For instance, due to the distributed nature of content availability in ICN, securing the content itself is much more important than securing the infrastructure or the end points. This lack of addressing security goals in the final ICN paradigm is even more critical when considering the coexistence of TCP/IP and ICN, which could lead to the introduction of new attacks and security

issues. One of the main limitations of existing projects is that all of them address only the existence of a transition phase without investigating the impact of coexistence on the security and privacy of the system. We believe that not only passing through this intermediate step is unavoidable, but also that it is important to assess the security and privacy vulnerabilities that might come up under the coexistence of both architectures.

- *Coexistence solutions that preserve inherent ICN advantages:* Due to its inherent features such as in-network caching, interest aggregation, and content oriented security, ICN provides improved communication system and security by design. Therefore, these essential features of ICN should be protected while designing a coexistence architecture.

- *Optimized ICN-IP name-space mapping:* An important issue in the state-of-the-art solutions that provide translation of IP/HTTP-level services into ICN (or vice versa) is to ensure that the communication latency is comparable with the one in the current networks. In most of the coexistence solutions, that use some sort of translation at any networking layer (e.g., transport or network), the main problem is the repeated sending of newly published content information towards the translation server, which generates delay in the response path of requester and congestion in the network. The problem lies in the fact that the URL is likely different for every request (assuming some form of meaningful service interaction between IP client and ICN producer). Additionally, the existing channel semantics cannot be applied directly because the corresponding routing identifier at the ICN level is different for each publication, from the translation server to IP client. Also, realizing the rendezvous function (RVZ) approach, which is responsible for the response of new publications, requires continue interaction between server and RZV. This causes an additional latency for the client requests, waiting for a fresh mapping of ICN-IP at each published event.

- *Data protection and confidentiality:* Ensuring privacy for network entities (e.g., consumer and producer) in coexistence architecture is not a trivial task, mainly due to the poor privacy support provided in ICN. Hence, it is important to investigate how the privacy issues were dealt in the current coexistence architectures. Ideally, names should reveal no more than what is currently revealed by an IP address and port. However, in ICN the name prefix reveals some information about the

content, and the in-network caching and data in PIT might expose the consumer identity. Therefore, the researchers should focus on the specific issues concerning the privacy and data protection in the co-existence scenarios. For instance, in a coexistence architecture, IP to name-prefix mapping is performed when an IP packet travels from IP to ICN network. In this scenario, the IP header does not reveal any information about the payload, but the prefix name does, thus, the data confidentiality is threatened when these data packets are traveling through the ICN "island". In particular, since the use of name prefix for addressing the data in ICN reveals sufficient information to the passive eavesdropper, ensuring privacy means that names and payloads cannot be correlated. However, such privacy requirement would need an upper-layer service similar to the one that would resolve non-topological identifiers (e.g., ICN name prefix) to topological names (e.g., IP network address).

- *SDN/NFV for efficient coexistence:* As mentioned earlier, the SDN technology separates the control plane from the data plane. The decoupled control plane is programmable and has a global view of the network that provides easier network management monitoring. SDN-based implementations of ICN exploit the centralized view available to the SDN controller, which enables the SDN controller to install appropriate rules in the data-plane to process ICN requests/responses. In the state-of-the-art, both *overlay* and *hybrid* ICN deployments have leveraged SDN to address different coexistence requirements, e.g., forwarding, storage, management, security, and interoperability. SDN has already been successfully adopted for network deployment; it makes SDN an appropriate choice for quick deployment of ICN with low hardware modifications. On the another side, NFV can help to virtualize several network functions that were previously implemented via physical devices.

6.5 Summary

In this chapter, we survey various efforts done by researchers and industries in recent years to propose a design of ICN-IP coexistence architecture. All of these architectures differ from each other according to their specific design, but they all adhere to the ICN paradigm, which means a content-oriented communication model in replacement of the current host-centric one. In our survey, we identify that all these architectures have important limita-

tions: none of them has been designed through a comprehensive approach that considers all the new challenges introduced by a coexistence scenario. Instead, the main aim for most of them is to improve the current Internet by exploiting some of the core ICN features (i.e., forwarding, storage, management, and security). Even though security also belongs to that list of features, none of the existing architectures has considered it as the main purpose. In future we believe appropriate coexistence architecture designs are needed to build and illuminate the secure path towards the future Internet (i.e., ICN). This can be done by considering limitations and necessary improvements of the existing coexistence solutions that we have suggested in this survey to design a complete and secure coexistence architecture. With the set of future research directions and open questions that we have raised, we hope that our work will motivate fledgling researchers towards designing a complete solution for ICN-IP coexistence while tackling the key security and privacy issues.

Chapter 7

Conclusion

The distinct features of Information Centric Networking paradigm such as receiver-driven mechanism, in-network caching, inherent support for mobility, and multi-cast routing makes it a perfect candidate to fit in the design space Future Internet Architecture. Numerous ICN styles have gained considerable attention in research groups of both academia and industry, and exposed the importance of ICN as a valuable alternative to current TCP/IP-based Internet.

To be a viable Internet architecture, ICN must be resilient against current and emerging threats. In this chapter we summarize what is our contribution in the securing ICN and we propose some directions for future works.

7.1 Contributions

The main contributions of this dissertation focus on securing ICN architectures. In particular, we addressed the following topics:

- **Secure and Efficient Adaptive Multimedia Streaming in ICN.** The key features of ICN which includes in-network caching and native tendency to support multicast has been shown to have unforeseen security consequences [18]. In this topic, first we show that how an adversary is able to exploit the implicit features of ICN (i.e., in-network caching and interest aggregation) and the adaptive streaming mechanism of Dynamic Adaptive Streaming (DAS), to degrade the performance of DAS over ICN. In particular, we show that the adversary

is able to harm the adaptive behaviour of DAS streaming control system, which leads to the degradation of user perceived QoE. We believe that our proposed so-called Bitrate Oscillation Attack (BOA), which is supported by a comprehensive investigation is essential before ICN can be considered adequate for DAS, and it is deployed for real-world multimedia applications.

To mitigate BOA, we proposed two counter solutions. First, we proposed a receiver driven solution called *Fair-RTT-DAS* to countermeasure the attack. Fair-RTT-DAS is a scalable adaptive rate control technique which enhances user perceived QoE in the presence of an adversary and ICN vital features. In particular, Fair-RTT-DAS uses the concept of maintaining RTT and throughput fairness in ICN's dynamic network condition to alleviate the adverse effects of adversary. Moreover, it shows that it can further enhance the user perceived QoE in presence of varied content source locations and ICN's implicit characteristics. In the second solution, we proposed a countermeasure, called CoMon-DAS, which eliminates the deficiencies of ICN architecture and protect the network against BOA. CoMon-DAS alleviates the effects of adversary by enabling network-wide coordinated caching and cache-aware routing in ICN. We validated our proposed approaches (the BOA and its countermeasures) via extensive simulations that are done on AMuSt-ndnSIM. Through result evaluation and analysis, we conclude that: (i) high frequency of bitrate switching increases the annoyance factor in spatial dimension, (ii) high amplitude of oscillations decreases the satisfactory visual quality, and (iii) mitigation approaches can significantly enhance the perceived QoE in presence of varied content source locations and BOA. The results of our contribution under this topic appeared in [54, 55, 97]

- **Authentication protocol for ICN based Mobile Networks.** Under this topic, we exploited the ICN's intrinsic mobility support and communication model to offer security services to higher network layers. In particular, we proposed a simplified LTE architecture that does not require the Mobility Management Entity (MME); i.e., the entity that guarantees an uninterrupted device connection during mobility events. We use the ICN communication style to design a revised device authentication protocol and a novel handover authentication protocol that reduced the number of exchanged messages between the authenticating entities. Our proposed handover authentication mechanism exploits the ICN synchronization protocol to securely transfer the

device security context (i.e., cryptographic material established during the mobile device authentication) during the handover mechanism from the previous to new base station.

Our approach reduced the complexity of the LTE infrastructure thus making it simpler, easier to manage and more cost-effective for network providers. We believe that this is a valid reason that would lead network providers for deploying ICN in their cellular infrastructure. The results of this topic appeared in [53].

- **Secure Mobility Management in ICN.** In the context of ICN, the practice of producer/consumer communication model primarily appreciates seamless mobility support to mobile devices. However, the dynamic interaction between producer and network's forwarding plane in ICN introduces new security challenges in the network. In this topic, we investigated and proposed solutions to the security challenges related to the producer mobility tracing-based protocols. Particularly, to mitigate prefix hijacking attacks and resolve security and privacy issues in ICN mobility management, we presented an efficient blockchain based distributed prefix authentication protocol, which offers reliable, secure and faster mobile producer authentication. We show that our proposed protocol is completely distributed and lightweight which can be easily deployed on different network access platforms (e.g., 4G, 5G, and WiFi). The security and performance analysis shows that our proposed protocol performs significantly better when compared to state-of-the-art, and it efficiently mitigates prefix hijacking, Denial of Service (DoS), and other telecommunication networking related attacks. The results of our contribution under this topic appeared in [59].

- **Denial of Service Mitigation in ICN.** In this topic, we address to architectural security issues that are intrinsic to the ICN design. particularly, where attackers exploit the new structure introduced by ICN, i.e, Pending Interest Table (PIT), to build a new type of Distributed DoS, better known as Interest Flooding Attack (IFA). We proposed a novel mechanism for IFA detection and mitigation, aimed at reducing the memory consumption of the PIT by effectively reducing the malicious traffic that passes through each ICN router. We evaluate the effectiveness our proposed mechanism through extensive simulations on the ndnSIM simulator and compared its effectiveness with the one achieved by the existing state-of-the-art. The results show that our proposed protocol effectively reduces the IFA damages, especially on

the legitimate traffic, with improvements with respect to the existing state-of-the-art. The results of this topic appeared in [35]

- **A Survey on ICN-IP Coexistence Solutions**

  In this topic, we survey various efforts done by researchers and industries in recent years which proposed the designs of ICN-IP coexistence architectures. In our survey, we provide a comprehensive analysis, classification and comparison of all the main existing coexistence architectures under set of relent features which are necessary for accurate ICN-IP coexistence. Through a thorough analysis, we identified the open issues and challenges present in existing architectures which are addressing ICN-IP coexistence. We also discussed some possible insights to address these issues. We believe that our work will motivate researchers towards designing a complete solution for ICN-IP coexistence while tackling the key security and privacy issues. Our contribution under this topic appeared in [56].

7.2 Future Directions

Based on the given dissertation, in the following we outline some directions for future works:

- **Efficient Adaptive Multimedia Streaming.** ICN's support for in-network caching and multicast are beneficial for the multimedia content provider in terms of lower transmission delay and reduced bandwidth requirements. However, in [54, 55], we have already shown that it makes Dynamic Adaptive Streaming (DAS) to be more challenging in ICN by exposing it to new security risks. Adaptive video streaming applications such as DAS requires a moderately high and also stable amount of bandwidth between the video sources and the clients, to avoid quality fluctuation and gain high Quality of Experience (QoE). Thus, the availability of multimedia data through multiple paths (i.e., due to in-network caching and interest aggregation) to connect the clients to the sources is not actually advantageous. This is because inherent characteristics of ICN are ignorant of multimedia characteristics, e.g., resolution, codec, bitrate. On the other hand, without such coordination, requests for similar multimedia data (and its qualities) also raise cache redundancy in the network which wastes network resources as well.

One proposal to make ICN network aware of multimedia data and DAS characteristics is by enabling network coding [178] within the ICN architecture. For efficient multipath streaming in ICN, both the network and multimedia data should make coordination among data and DAS characteristics. We believe that a deep investigation is required on identifying such characteristics and applying the coordination between them to provide the best possible perceived QoE. In addition, we plan to investigate in the direction of making caching decisions more intelligent. For example, by enabling the network to convert high-resolution segments to lower ones (at line rate), it becomes possible to cache the highest available qualities and dynamically transforming them (if necessary) to provide the best possible QoE. As for the coordination framework, we plan to redesign the domain controller in a distributed way, both to make it more tolerant to faults and also for load balancing.

- **Security Services for upcoming Technologies.**

  The so-called Internet of Things (IoT) devices which also called as "smart" are pouring into our environments. These resource-constraint devices, often employed as a group (known as "swarm") in different applications (e.g., smart cities, smart factories, oil and gas sector, biomedical). Incidents like Stuxnet[1] proves the necessity of the security of these devices. As these devices perform a crucial task, it is indeed critical to maintain their integrity. In the past decade, remote attestation (RA) has emerged as a distinct solution to detect malware presence on these devices over the network. More recently, researchers have started exploring efficient ways to perform remote attestation over swarms. Unfortunately those developed methods [27, 31, 99] could not provide solution for device mobility during attestation. We believe that ignoring device mobility and enforcing static nature during attestation is a weakness in proposed RA schemes. ICN enables mobility management completely at the network layer. Leveraging ICN's implicit features to provide remote attestation for IoT devices in dynamic network conditions is an interesting topic to investigate.

- **Denial-of-Service.** Denial-of-Service attacks are difficult to prevent and to make them harmless. ICNs specific DoS attacks, such as Interest Flooding attacks (IFA), are not an exception.

[1] https://www.wired.com/2014/11/countdown-to-zero-day-stuxnet

Several detection and reaction mechanisms have been trying to mitigate IFA [19,49,60], but all of them are not highly effective since most of them focus on reducing PIT size just by focusing on incoming traffic drop, thus heavily damages the legitimate traffic. In this dissertation, we take a footstep in the direction of identifying and differentiating malicious packets from the benign traffic during IFA. In particular, the proposed protocol differentially penalizes malicious traffic by identifying malicious traffic flow. Although the results of our evaluation show that proposed protocol is able to provide a high quality of service to legitimate traffic, still the intuition of protocol is to stabilize router's PIT up to certain minimum level during the attack. In the future, we will investigate to improve the network health by aiming to make PIT unaffected during the attack, and keeping the efficacy of the proposed protocol.

- **User privacy.** Another key subject which requires consideration of researchers is the user privacy in ICN. In this thesis, we showed that adversary which is able to exploit timing attacks [18] as a side channel to breach users privacy can infer the content which has been previously (or currently) requested by the user. In particular, by probing the MPD files and exploiting the timing attacks adversary is able to predict the video that the user is going to stream. In [54], we already showed that this privacy breach further allows the adversary to adversely exploit in-network caching and interest collapsing features to launch the Bitrate Oscillation Attack (BOA). An example to mitigate such privacy breach is presented in [54], where privacy aware forwarding [18] is implemented by adding content-specific delay on each cache hit. Such a method can be applied to mitigate the BOA, however, its adoption would make impossible to exploit interest collapsing and in-network caching, thus reducing ICN performance. We believe that privacy in ICN justifies a deeper investigation to evade the choice between privacy and performance improvement in communication.

Bibliography

[1] architectuRe for an Internet For Everybody (RIFE). https://rife-project.eu/.

[2] DeplOyment and seCurisaTion of new functiOnalities in virtualized networking enviRonments (DOCTOR). http://www.doctor-project.org/.

[3] General Packet Radio Services (GPRS) service description stage. *3GPP TS 33.060*.

[4] Information-Centric Networking Research Group (ICNRG). https://irtf.org/icnrg.

[5] NSF Future Internet Architecture Project. http://www.nets-fia.net/.

[6] POINT (iP Over IcN the betTer IP). https://www.point-h2020.eu/.

[7] Scalable Adaptive Internet soLutions (SAIL) European Commissions 7th Framework Program. http://www.sail-project.eu/about-sail/index.html.

[8] Internet Protocol. RFC 791, September 1981.

[9] Transmission Control Protocol. RFC 793, September 1981.

[10] Domain names - concepts and facilities. RFC 1034, November 1987.

[11] Domain names - implementation and specification. RFC 1035, November 1987.

[12] A survey of BGP security issues and solutions. *Proceedings of the IEEE*, 98(1):100–122, 1 2010.

[13] An ANFIS-based cache replacement method for mitigating cache pollution attacks in Named Data Networking. *Computer Networks*, 80:51 – 65, 2015.

[14] ETSI TS 102 165-1 V4.2.3 (2011-03). Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Methods and protocols; Part 1: Method and proforma for Threat, Risk, Vulnerability Analysis, 2011.

[15] 3GPP Consortium. http://www.3gpp.org/.

[16] 3rd Generation Partnership Project. 3GPP System Architecture Evolution (SAE). In *"Technical Specification Group Services and System Aspects"*, 2008.

[17] Ouaddah Aafaf, Abou Elkalam Anas, and Ait Ouahman Abdellah. FairAccess: a new Blockchain-based access control framework for the Internet of Things. *Security and Communication Networks*, 9(18):5943–5964.

[18] G. Acs, M. Conti, P. Gasti, C. Ghali, G. Tsudik, and C. Wood. Privacy-Aware Caching in Information-Centric Networking. *IEEE Transactions on Dependable and Secure Computing*, PP(99), 2017.

[19] Alexander Afanasyev, Priya Mahadevan, Ilya Moiseenko, Ersin Uzun, and Lixia Zhang. Interest flooding attack and countermeasures in Named Data Networking. In *IFIP Networking Conference, 2013*, pages 1–9. IEEE, 2013.

[20] Alexander Afanasyev, Ilya Moiseenko, and Lixia Zhang. ndnSIM: NDN simulator for NS-3. Technical Report NDN-0005, NDN, October 2012.

[21] Alexander Afanasyev, Ilya Moiseenko, Lixia Zhang, et al. ndnSIM: NDN simulator for NS-3. *University of California, Los Angeles, Tech. Rep*, 4, 2012.

[22] Suvrat Agrawal, Samar Shailendra, Bighnaraj Panigrahi, Hemant Kumar Rath, and Anantha Simha. O-ICN Simulator (OICNSIM): An

NS-3 Based Simulator for Overlay Information Centric Networking (O-ICN). In *Proceedings of the 1st Workshop on Complex Networked Systems for Smart Infrastructure*, CNetSys '18, pages 13–15, New York, NY, USA, 2018. ACM.

[23] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman. A survey of information-Centric networking. *IEEE Communications Magazine*, 50(7):26–36, July 2012.

[24] Saamer Akhshabi, Sethumadhavan Narayanaswamy, Ali C. Begen, and Constantine Dovrolis. An Experimental Evaluation of Rate-adaptive Video Players over HTTP. *Image Commun.*, 27(4):271–287, April 2012.

[25] Mohammad Al-Shurman, Seong-Moo Yoo, and Seungjin Park. Black Hole Attack in Mobile Ad Hoc Networks. In *Proceedings of the 42Nd Annual Southeast Regional Conference*, ACM-SE 42, pages 96–97, New York, NY, USA, 2004. ACM.

[26] K. A. Alezabi et al. An efficient authentication and key agreement protocol for 4G (LTE) networks. In *IEEE Region 10 Symposium*, pages 502–507, 2014.

[27] Moreno Ambrosin, Mauro Conti, Ahmad Ibrahim, Gregory Neven, Ahmad-Reza Sadeghi, and Matthias Schunter. SANA: Secure and Scalable Aggregate Network Attestation. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 731–742, 2016.

[28] Carlos Anastasiades, Torsten Braun, and Vasilios A. Siris. *Information-Centric Networking in Mobile and Opportunistic Networks*, pages 14–30. Springer International Publishing, Cham, 2014.

[29] Somaya Arianfar, Pekka Nikander, and Jörg Ott. On Content-centric Router Design and Implications. In *Proceedings of the Re-Architecting the Internet Workshop*, ReARCH '10, pages 5:1–5:6, New York, NY, USA, 2010. ACM.

[30] Jari Arkko, Vesa Lehtovirta, and Pasi Eronen. Improved Extensible Authentication Protocol method for 3rd generation Authentication and Key Agreement (EAP-AKA). 2009.

[31] N Asokan, Ferdinand Brasser, Ahmad Ibrahim, Ahmad-Reza Sadeghi, Matthias Schunter, Gene Tsudik, and Christian Wachsmann. SEDA:

Scalable embedded device attestation. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 964–975, 2015.

[32] Jordan Augé et al. Anchor-less Producer Mobility in ICN. In *ACM Proceedings of the 2nd International Conference on Information-Centric Networking*, pages 189–190, 2015.

[33] Hitesh Ballani, Paul Francis, and Xinyang Zhang. A Study of Prefix Hijacking and Interception in the Internet. *SIGCOMM Comput. Commun. Rev.*, 37(4):265–276, August 2007.

[34] Harkeerat Bedi, Sankardas Roy, and Sajjan Shiva. Mitigating congestion based DoS attacks with an enhanced AQM technique. *Computer Communications*, 56:60 – 73, 2015. "http://www.sciencedirect.com/science/article/pii/S0140366414003107".

[35] Abdelmadjid Benarfa, Muhammad Hassan, Mohamed bachir, Alberto Compagno, Eleonora Losiouk, and Mauro Conti. Choose to Kill IFA (ChoKIFA): A New Detection and Mitigation Approach against Interest Flooding Attacks in NDN (Under submission). *In Proceedings of the 17th International Conference on Wired/Wireless Internet Communications (IFIP WWIC), Bologna, Italy*, June 17-18, 2019.

[36] Daniel J. Bernstein and Tanja Lange. eBACS: ECRYPT Benchmarking of Cryptographic Systems. https://bench.cr.yp.to/.

[37] Giacomo Brambilla, Michele Amoretti, and Francesco Zanichelli. Using Block Chain for Peer-to-Peer Proof-of-Location. *CoRR*, abs/1607.00174, 2016.

[38] N. Brownlee, C. Mills, and G. Ruth. Traffic Flow Measurement: Architecture, 1997.

[39] J. Calvert. The electromagnetic telegraph, 2008. http://mysite.du.edu/~jcalvert/tel/telhom.htm.

[40] A. T. Campbell, J. Gomez, S. Kim, A. G. Valko, Chieh-Yih Wan, and Z. R. Turanyi. Design, implementation, and evaluation of cellular IP. *IEEE Personal Communications*, 7(4):42–49, Aug 2000.

[41] J. Cao, M. Ma, H. Li, Y. Zhang, and Z. Luo. A Survey on Security Aspects for LTE and LTE-A Networks. *IEEE COMST*, 16:283–302, 2014.

[42] G. Carofiglio, M. Gallo, and L. Muscariello. ICP: Design and evaluation of an Interest control protocol for content-centric networking. In *2012 Proceedings IEEE INFOCOM Workshops*, pages 304–309, March 2012.

[43] G. Carofiglio, M. Gallo, L. Muscariello, and M. Papalini and. Optimal multipath congestion control and request forwarding in Information-Centric Networks. In *2013 21st IEEE International Conference on Network Protocols (ICNP)*, pages 1–10, Oct 2013.

[44] J. Chan et al. A practical user mobility prediction algorithm for supporting adaptive QoS in wireless networks. In *IEEE International Conference on Networks*, pages 104–111, 1999.

[45] M. Chatterjee, S. K. Das, and D. Turgut. An on-demand weighted clustering algorithm (WCA) for ad hoc networks. In *Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE*, volume 3, pages 1697–1701 vol.3, Nov 2000.

[46] D.R. Cheriton and M. Gritter. TRIAD: A New Next-Generation Internet Architecture, 2000.

[47] Parminder Chhabra, Shobhit Chuig, Anurag Goel, Ajita John, Abhishek Kumar, Huzur Saran, and Rajeev Shorey. XCHOKe: Malicious source control for congestion avoidance at Internet gateways. In *Network Protocols, 2002. Proceedings. 10th IEEE International Conference on*, pages 186–187. IEEE, 2002.

[48] Dave Clark, Lehr, Steve Bauer, Peyman Faratin, Rahul Sami, and John Wroclawski. The Growth of Internet Overlay Networks : Implications for Architecture , Industry Structure and Policy. 2005.

[49] Alberto Compagno, Marco Conti, Paolo Gasti, and Gene Tsudik. Poseidon: Mitigating interest flooding DDoS attacks in named data networking. In *Local Computer Networks (LCN), 2013 IEEE 38th Conference on*, pages 630–638. IEEE, 2013.

[50] Alberto Compagno, Mauro Conti, Paolo Gasti, Luigi Vincenzo Mancini, and Gene Tsudik. *Violating Consumer Anonymity: Geo-Locating Nodes in Named Data Networking*, pages 243–262. Springer International Publishing, Cham, 2015.

[51] Alberto Compagno, Mauro Conti, and Muhammad Hassan. *An ICN-Based Authentication Protocol for a Simplified LTE Architecture*. Springer International Publishing, Cham, 2018.

[52] Alberto Compagno et al. OnboardICNg: A Secure Protocol for Onboarding IoT Devices in ICN. In *ACM Conference on Information-Centric Networking*, pages 166–175, 2016.

[53] Alberto Compagno, Xuan Zeng, Luca Muscariello, Giovanna Carofiglio, and Jordan Augé. Secure Producer Mobility in Information-Centric Network. In *Proceedings of the 4th ACM Conference on Information-Centric Networking*, ICN '17, pages 163–169, New York, NY, USA, 2017. ACM.

[54] M. Conti, R. Droms, M. Hassan, and S. Valle. QoE Degradation Attack in Dynamic Adaptive Streaming over ICN. In *IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2018.

[55] Mauro Conti, Ralph Droms, Muhammad Hassan, and Chhagan Lal. Fair-RTT-DAS: A Robust and Efficient Dynamic Adaptive Streaming over ICN. *Computer Communications*, 129:209 – 225, 2018.

[56] Mauro Conti, Ankit Gangwal, Muhammad Hassan, Chaggan Lal, and Eleonora Losiouk. The Road Ahead for Networking: A Survey on ICN-IP Coexistence Solutions. *IEEE Communications Surveys and Tutorials (COMST), 2019 (Under submission).*

[57] Mauro Conti, Paolo Gasti, and Marco Teoli. A Lightweight Mechanism for Detection of Cache Pollution Attacks in Named Data Networking. *Comput. Netw.*, 57(16):3178–3191, November 2013.

[58] Mauro Conti, Paolo Gasti, and Marco Teoli. A Lightweight Mechanism for Detection of Cache Pollution Attacks in Named Data Networking. volume 57, pages 3178–3191, New York, NY, USA, November 2013. Elsevier North-Holland, Inc.

[59] Mauro Conti, Muhammad Hassan, and Chaggan Lal. BlockAuth: BlockChain based Distributed Producer Authentication in ICN. *Elsevier Computer Communications, 2018 (Under submission).*

[60] Huichen Dai, Yi Wang, Jindou Fan, and Bin Liu. Mitigate ddos attacks in ndn by interest traceback. In *Computer Communications*

Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on, pages 381–386. IEEE, 2013.

[61] Christian Dannewitz and et al. Network of Information (NetInf) - An Information-centric Networking Architecture. *Elsevier Computer Communication*, 36(7):721–735, 2013.

[62] S. Das, A. Misra, and P. Agrawal. TeleMIP: telecommunications-enhanced mobile IP architecture for fast intradomain mobility. *IEEE Personal Communications*, 7(4):50–58, Aug 2000.

[63] Subir Das et al. TeleMIP: telecommunications-enhanced mobile IP architecture for fast intradomain mobility. *IEEE Personal Communications*, 7(4):50–58, 2000.

[64] A. Detti, M. Pomposini, N. Blefari-Melazzi, S. Salsano, and A. Bragagnini. Offloading cellular networks with Information-Centric Networking: The case of video streaming. In *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–3, June 2012.

[65] Andrea Detti and et al. CONET: A Content Centric Inter-networking Architecture. In *ACM SIGCOMM workshop on Information Centric networking*, pages 50–55, 2011.

[66] Mohamed Diallo, Serge Fdida, Vasilis Sourlas, Paris Flegkas, and Leandros Tassiulas. Leveraging Caching for Internet-Scale Content-Based Publish/Subscribe Networks. *2011 IEEE International Conference on Communications (ICC)*, pages 1–5, 2011.

[67] Vladimir Dimitrov and et al. PSIRP Project – Publish-subscribe Internet Routing Paradigm: New Ideas for Future Internet. In *11th ACM CompSysTech*, pages 167–171, 2010.

[68] Konstantinos D. Dimou, Min Wang, Yu Yang, Muhammad Kazmi, Anna Larmo, Jonas Pettersson, Walter Mller, and Ylva Timner. Handover within 3GPP LTE: Design Principles and Performance. In *VTC Fall*, pages 0–. IEEE, 2009.

[69] A. Dorri, S. S. Kanhere, and R. Jurdak. Towards an Optimized BlockChain for IoT. In *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 173–178, April 2017.

[70] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak. BlockChain: A Distributed Solution to Automotive Security and Privacy. *IEEE Communications Magazine*, 55(12):119–125, DECEMBER 2017.

[71] Ali Dorri, Salil S. Kanhere, Raja Jurdak, and Praveen Gauravaram. LSB: A Lightweight Scalable BlockChain for IoT Security and Privacy. *CoRR*, abs/1712.02969, 2017.

[72] Chethana Dukkipati, Yunpeng Zhang, and Liang Chieh Cheng. Decentralized, BlockChain Based Access Control Framework for the Heterogeneous Internet of Things. In *Proceedings of the Third ACM Workshop on Attribute-Based Access Control*, ABAC'18, pages 61–69, New York, NY, USA, 2018. ACM.

[73] Kjeld Borch Egevang and Pyda Srisuresh. Traditional IP Network Address Translator (Traditional NAT). RFC 3022, January 2001.

[74] L. Zhang et al. Named Data Networking (NDN) Project.

[75] Kevin Fall. A Delay-tolerant Network Architecture for Challenged Internets. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '03, pages 27–34, New York, NY, USA, 2003. ACM.

[76] 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*, pages 1–1, 2018.

[77] Hamid Farhady, HyunYong Lee, and Akihiro Nakao. Software Defined Networking: A Survey. *Elsevier Computer Networks*, 81:79–95, 2015.

[78] Wu-chang Feng, Dilip D Kandlur, Debanjan Saha, and Kang G Shin. Stochastic fair blue: A queue management algorithm for enforcing fairness. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1520–1529. IEEE, 2001.

[79] Wu-chang Feng, Kang G Shin, Dilip D Kandlur, and Debanjan Saha. The BLUE active queue management algorithms. *IEEE/ACM Transactions on Networking (ToN)*, 10(4):513–528, 2002.

[80] Eduardo Castelló Ferrer. The blockchain: a new framework for robotic swarm systems. *CoRR*, abs/1608.00695, 2016.

[81] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on networking*, 1(4):397–413, 1993.

[82] N. Fotiou and G. C. Polyzos. Decentralized name-based security for content distribution using blockchains. In *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 415–420, April 2016.

[83] Conner Fromknecht, Dragos Velicanu, and Sophia Yakoubov. A Decentralized Public Key Infrastructure with Identity Retention. *IACR Cryptology ePrint Archive*, 2014:803, 2014.

[84] X. Fu, D. Kutscher, S. Misra, and R. Li. Information-Centric Networking Security. *IEEE Communications Magazine*, 56(11):60–61, November 2018.

[85] R. H. Garner. Forerunner in wireless telegraphy. *Journal of the Institution of Electrical Engineers*, 4(48):663–664, December 1958.

[86] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang. DoS and DDoS in Named Data Networking. In *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*, pages 1–7, July 2013.

[87] Paolo Gasti, Gene Tsudik, Ersin Uzun, and Lixia Zhang. DoS and DDoS in named data networking. In *Computer Communications and Networks (ICCCN), 2013 22nd International Conference on*, pages 1–7. IEEE, 2013.

[88] C. Ghali, A. Narayanan, D. Oran, G. Tsudik, and C. A. Wood. Secure Fragmentation for Content-Centric Networks. In *2015 IEEE 14th International Symposium on Network Computing and Applications*, pages 47–56, Sept 2015.

[89] Cesar Ghali. Needle in a Haystack : Mitigating Content Poisoning in Named-Data Networking. 2014.

[90] Cesar Ghali, Marc A. Schlosberg, Gene Tsudik, and Christopher A. Wood. Interest-Based Access Control for Content Centric Networks. In *Proceedings of the 2Nd ACM Conference on Information-Centric Networking*, ACM-ICN '15, pages 147–156, New York, NY, USA, 2015. ACM.

[91] M. E. Gorman and Alexander graham bell. Encyclopedia of Creativity, Two-Volume Set, 1999.

[92] Visvasuresh Victor Govindaswamy, Gergely Záruba, and G Balasekaran. RECHOKe: a scheme for detection, control and punishment of malicious flows in IP networks. In *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*, pages 16–21. IEEE, 2007.

[93] Peter Gusev and Jeff Burke. NDN-RTC: Real-Time Videoconferencing over Named Data Networking. In *Proceedings of the 2Nd ACM Conference on Information-Centric Networking*, ACM-ICN '15, pages 117–126, New York, NY, USA, 2015. ACM.

[94] M. Hajjar, G. Aldabbagh, and N. Dimitriou. Using clustering techniques to improve capacity of LTE networks. In *2015 21st Asia-Pacific Conference on Communications (APCC)*, pages 68–73, Oct 2015.

[95] Dookyoon Han, Munyoung Lee, Kideok Cho, T. Kwon, and Y. Choi. Publisher mobility support in content centric networks. In *The International Conference on Information Networking 2014 (ICOIN2014)*, pages 214–219, Feb 2014.

[96] Sayed Hadi Hashemi, Faraz Faghri, and Roy H. Campbell. Decentralized User-Centric Access Control using PubSub over Blockchain. *CoRR*, abs/1710.00110, 2017.

[97] Muhammad Hassan, Hani Salah, Mauro Conti, Frank H. P. Fitzek, and Thorsten Sturfe. CoMon-DAS: A Framework for Efficient and Robust Dynamic Adaptive Streaming over NDN. *In proceedings of 24th IEEE Symposium on Computers and Communications (ISCC), in press, Barcelona, Spain*, June 29 - July 3, 2019.

[98] G. Huston, M. Rossi, and G. Armitage. Securing BGP x2014; A Literature Survey. *IEEE Communications Surveys Tutorials*, 13(2):199–222, Second 2011.

[99] Ahmad Ibrahim, Ahmad-Reza Sadeghi, Gene Tsudik, and Shaza Zeitouni. DARPA: Device attestation resilient to physical attacks. In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, WiSec '16, pages 171–182, 2016.

[100] Y. E. H. E. Idrissi et al. Security analysis of 3GPP (LTE) - WLAN interworking and a new local authentication method based on EAP-

AKA. In *International Conference on Future Generation Communication Technologies*, pages 137–142, 2012.

[101] Van Jacobson et al. Networking Named Content. In *ACM International Conference on Emerging Networking Experiments and Technologies*, pages 1–12, 2009.

[102] J. Jiang, V. Sekar, and H. Zhang. Improving Fairness, Efficiency, and Stability in HTTP-Based Adaptive Video Streaming With Festive. *IEEE/ACM Transactions on Networking*, 22(1):326–340, Feb 2014.

[103] X. Jiang, J. Yang, G. Jin, and W. Wei. RED-FT: A Scalable Random Early Detection Scheme with Flow Trust against DoS Attacks. *IEEE Communications Letters*, 17(5):1032–1035, May 2013.

[104] Petri Jokela, András Zahemszky, Christian Esteve Rothenberg, Somaya Arianfar, and Pekka Nikander. LIPSIN: line speed publish/subscribe inter-networking. In *SIGCOMM*, 2009.

[105] S. Kent, C. Lynn, and K. Seo. Secure Border Gateway Protocol (S-BGP). *IEEE Journal on Selected Areas in Communications*, 18(4):582–592, April 2000.

[106] Stephen Kent. IP Authentication Header. RFC 4302, December 2005.

[107] Stephen Kent. IP Encapsulating Security Payload (ESP). RFC 4303, December 2005.

[108] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz. Disruption-Tolerant Networking: A Comprehensive Survey on Recent Developments and Persisting Challenges. *IEEE Communications Surveys Tutorials*, 14(2):607–640, Second 2012.

[109] Do-hyung Kim, Jong-hwan Kim, Yu-sung Kim, Hyun-soo Yoon, and Ikjun Yeom. Mobility Support in Content Centric Networks. In *Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking*, ICN '12, pages 13–18, New York, NY, USA, 2012. ACM.

[110] Tero Kivinen, Ari Huttunen, Brian Swander, and Victor Volpe. Negotiation of NAT-Traversal in the IKE. RFC 3947, January 2005.

[111] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M. Frans Kaashoek. The Click Modular Router. *ACM Trans. Comput. Syst.*, 18(3):263–297, August 2000.

[112] Teemu Koponen, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, and Ion Stoica. A Data-oriented (and Beyond) Network Architecture. In *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '07, pages 181–192, New York, NY, USA, 2007. ACM.

[113] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 839–858, May 2016.

[114] Diego Kreutz, Fernando MV Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. Software Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*, 103(1):14–76, 2015.

[115] Christian Kreuzberger, Daniel Posch, and Hermann Hellwagner. A Scalable Video Coding Dataset and Toolchain for Dynamic Adaptive Streaming over HTTP. In Tsang Ooi Wei, editor, *Proceedings of the 6th ACM Multimedia Systems Conference*, pages 213–218, New York, NY, USA, mar 2015. ACM.

[116] Christian Kreuzberger, Daniel Posch, and Hermann Hellwagner. AMuSt Framework - Adaptive Multimedia Streaming Simulation Framework for ns-3 and ndnSIM, 2016.

[117] Srisankar S Kunniyur and Rayadurgam Srikant. An adaptive virtual queue (AVQ) algorithm for active queue management. *IEEE/ACM Transactions on Networking (ToN)*, 12(2):286–299, 2004.

[118] James F. Kurose and Keith Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2002.

[119] Hyeyeon Kwon et al. Handover prediction strategy for 3G-WLAN overlay networks. In *IEEE Network Operations and Management*, pages 819–822, 2008.

[120] G. M. Kien. Mutual entity authentication for LTE. In *International Wireless Communications and Mobile Computing Conference*, pages 689–694, 2011.

[121] Rami Langar, Nizar Bouabdallah, and Raouf Boutaba. Mobility-aware Clustering Algorithms with Interference Constraints in Wireless Mesh Networks. *Comput. Netw.*, 53(1):25–44, January 2009.

[122] Nikolaos Laoutaris, Hao Che, and Ioannis Stavrakakis. The LCD interconnection of LRU caches and its analysis. *Performance Evaluation*, 63(7):609 – 634, 2006.

[123] S. Lederer, C. Mueller, B. Rainer, C. Timmerer, and H. Hellwagner. Adaptive streaming over Content Centric Networks in mobile networks using multiple links. In *2013 IEEE International Conference on Communications Workshops (ICC)*, pages 677–681, June 2013.

[124] S. Lederer, C. Mueller, B. Rainer, C. Timmerer, and H. Hellwagner. An experimental analysis of Dynamic Adaptive Streaming over HTTP in Content Centric Networks. In *2013 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, July 2013.

[125] S. Lederer, C. Mueller, C. Timmerer, and H. Hellwagner. Adaptive multimedia streaming in information-Centric networks. *IEEE Network*, 28(6):91–96, Nov 2014.

[126] Stefan Lederer, Christopher Mueller, and Christian Timmerer. Dynamic Adaptive Streaming over HTTP Dataset. In Mark Claypool, Carsten Griwodz, and Ketan Mayer-Patel, editors, *Proceedings of the Third Annual ACM SIGMM Conference on Multimedia Systems (MMSys12)*, pages 89–94, New York, NY, USA, feb 2012. ACM.

[127] Stefan Lederer, Christopher Mueller, Christian Timmerer, Cyril Concolato, Jean Le Feuvre, and Karel Fliegel. Distributed DASH Dataset. In *Proceedings of the 4th ACM Multimedia Systems Conference*, MMSys '13, pages 131–135, New York, NY, USA, 2013. ACM.

[128] Stefan Lederer, Christopher Müller, and Christian Timmerer. Dynamic Adaptive Streaming over HTTP Dataset. In *Proceedings of the 3rd Multimedia Systems Conference*, MMSys '12, pages 89–94, New York, NY, USA, 2012. ACM.

[129] W. Li, S. Oteafy, and H. Hassanein. Rate-Selective Caching for Adaptive Streaming over Information-Centric Networks. *IEEE Transactions on Computers*, PP(99):1–1, 2017.

[130] Yong Li and Min Chen. Software Defined Network Function Virtualization: A Survey. *IEEE Access*, 3:2542–2553, 2015.

[131] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, and D. Oran. Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale. *IEEE Journal on Selected Areas in Communications*, 32(4):719–733, April 2014.

[132] Marc Liberatore and Brian Neil Levine. Inferring the Source of Encrypted HTTP Connections. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, CCS '06, pages 255–263, New York, NY, USA, 2006. ACM.

[133] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 15(7):1265–1275, Sep 1997.

[134] Dong Lin and Robert Morris. Dynamics of random early detection. In *ACM SIGCOMM Computer Communication Review*, volume 27, pages 127–137. ACM, 1997.

[135] Y. Liu, S. Dey, D. Gillies, F. Ulupinar, and M. Luby. User Experience Modeling for DASH Video. In *2013 20th International Packet Video Workshop*, pages 1–8, Dec 2013.

[136] Y. Liu, J. Geurts, J. C. Point, S. Lederer, B. Rainer, C. Mller, C. Timmerer, and H. Hellwagner. Dynamic adaptive streaming over CCN: A caching and overhead analysis. In *2013 IEEE International Conference on Communications (ICC)*, pages 3629–3633, June 2013.

[137] Y. Liu and J. Y. B. Lee. A unified framework for automatic quality-of-experience optimization in mobile video streaming. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, April 2016.

[138] A. Mason. CCSP Self-Study:. In *Cisco Secure Virtual Private Networks (CSVPN)*. Pearson Higher Education,, 2004.

[139] Spyridon Mastorakis, Alexander Afanasyev, Ilya Moiseenko, and Lixia Zhang. ndnSIM 2: An updated NDN simulator for NS-3. Technical Report NDN-0028, Revision 2, NDN, November 2016.

[140] Md Mehedi Masud. Survey of security features in LTE Handover Technology. *System*, 1:2, 2015.

[141] Nick McKeown, , and et al. Openflow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.

[142] Paolo Medagliani, Stefano Paris, Jeremie Leguay, Lorenzo Maggi, Chuangsong Xue, and Haojun Zhou. Overlay routing for fast video transfers in CDN. *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 531–536, 2017.

[143] N Blefari Melazzi and et al. An Openflow-based Testbed for Information Centric Networking. In *FutureNetw*, pages 1–9, 2012.

[144] Yang Ming. Analysis of physical-layer security in future mobile communication. In *Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC)*, pages 3144–3147, 2013.

[145] Axel Moinet, Benoît Darties, and Jean-Luc Baril. Blockchain based trust & authentication for decentralized sensor networks. *CoRR*, abs/1706.01730, 2017.

[146] Ilya Moiseenko and Dave Oran. TCP/ICN: Carrying TCP over Content Centric and Named Data Networks. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, ACM-ICN '16, pages 112–121, New York, NY, USA, 2016. ACM.

[147] Marc Mosko. CCNx 1.0 Protocol Specifications Roadmap.

[148] C. Mueller, S. Lederer, J. Poecher, and C. Timmerer. Demo paper: Libdash - An open source software library for the MPEG-DASH standard. In *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, July 2013.

[149] Sandra L. Murphy, Madelyn R. Badger, and Brian Wellington. OSPF with Digital Signatures. *RFC*, 2154:1–29, 1997.

[150] Luca Muscariello and et al. System and Method to Facilitate Integration of Information-centric Networking into Internet Protocol Networks. In *CISCO Technology, Inc.*, 2018. https://patents.google.com/patent/US8504609.

[151] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, http://bitcoin.org/bitcoin.pdf, 2008.

[152] John Naughton. The evolution of the Internet: from military experiment to General Purpose Technology. *Journal of Cyber Policy*, 1(1):5–28, 2016.

[153] Packet Fragmentation in NDN: Why NDN Uses Hop-By-Hop Fragmentation (NDN Memo), 2015. https://named-data.net/wp-content/uploads/2015/05/ndn-0032-1-ndn-memo-fragmentation.pdf.

[154] Pengpeng Ni, Ragnhild Eg, Alexander Eichhorn, Carsten Griwodz, and Pål Halvorsen. Flicker Effects in Adaptive Video Streaming to Handheld Devices. In *Proceedings of the 19th ACM International Conference on Multimedia*, MM '11, pages 463–472, New York, NY, USA, 2011. ACM.

[155] Y. Nicolas, D. Wolff, D. Rossi, and A. Finamore. I Tube, YouTube, P2PTube: Assessing ISP benefits of peer-assisted caching of YouTube content. In *IEEE P2P 2013 Proceedings*, pages 1–2, Sep. 2013.

[156] Ben Niven-Jenkins, Francois Le Faucheur, and Nabil Bitar. Content Distribution Network Interconnection (CDNI) Problem Statement. *RFC*, 6707:1–32, 2012.

[157] H. Orman. The Morris worm: a fifteen-year perspective. *IEEE Security Privacy*, 99(5):35–43, Sep. 2003.

[158] Jörg Ott and Dirk Kutscher. Why Seamless? Towards Exploiting WLAN-Based Intermittent Connectivity on the Road. In *TERENA Networking Conference*, 2004.

[159] S. Oueslati, J. Roberts, and N. Sbihi. Flow-aware traffic control for a content-centric network. In *2012 Proceedings IEEE INFOCOM*, pages 2417–2425, March 2012.

[160] Rong Pan, Balaji Prabhakar, and Konstantinos Psounis. CHOKe-a stateless active queue management scheme for approximating fair bandwidth allocation. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 942–951. IEEE, 2000.

[161] PARC. CCNx Over UDP.

[162] H. Park, I. Widjaja, and H. Lee. Detection of cache pollution attacks using randomness checks. In *2012 IEEE International Conference on Communications (ICC)*, pages 1096–1100, June 2012.

[163] Charles E Perkins. Mobile IP. *IEEE Communications Magazine*, 35(5):84–99, 1997.

[164] S. Petrangeli, N. Bouten, M. Claeys, and F. De Turck. Towards SVC-based Adaptive Streaming in information-Centric networks. In *2015 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 1–6, June 2015.

[165] G. P. Pollini. Trends in handover design. *IEEE Communications Magazine*, 34:82–90, 1996.

[166] I. Psaras, W. K. Chai, and G. Pavlou. In-Network Cache Management and Resource Allocation for Information-Centric Networks. *IEEE Transactions on Parallel and Distributed Systems*, 25(11):2920–2931, Nov 2014.

[167] M. Purkhiabani et al. Enhanced authentication and key agreement procedure of next generation evolved mobile networks. In *IEEE International Conference on Communication Software and Networks*, pages 557–563, 2011.

[168] Akbar Rahman, Dirk Trossen, Dirk Kutscher, and Ravi Ravindran. Deployment Considerations for Information-Centric Networking (ICN). Internet-Draft draft-irtf-icnrg-deployment-guidelines-04, Internet Engineering Task Force, September 2018. Work in Progress.

[169] B. Rainer, D. Posch, and H. Hellwagner. Investigating the Performance of Pull-Based Dynamic Adaptive Streaming in NDN. *IEEE Journal on Selected Areas in Communications*, 34(8):2130–2140, Aug 2016.

[170] Ravishankar Ravindran et al. Supporting seamless mobility in named data networking. In *IEEE International Conference on Communications*, pages 5854–5869, 2012.

[171] Jing Ren and et al. On the Deployment of Information-centric Network: Programmability and Virtualization. In *ICNC*, pages 690–694, 2015.

[172] H. Salah, J. Wulfheide, and T. Strufe. Coordination supports security: A new defence mechanism against interest flooding in NDN. In *2015 IEEE 40th Conference on Local Computer Networks (LCN)*, pages 73–81, Oct 2015.

[173] Hani Salah et al. Coordination supports security: A new defence mechanism against interest flooding in NDN. In *IEEE LCN*, 2015.

[174] Hani Salah et al. CoMon++: Preventing Cache Pollution in NDN Efficiently and Effectively. In *IEEE LCN*, 2017.

[175] Hani Salah and Thorsten Strufe. Comon: An Architecture for Coordinated Caching and Cache-aware Routing in CCN. In *IEEE CCNC*, 2015.

[176] Hani Salah and Thorsten Strufe. Evaluating and mitigating a collusive version of the interest flooding attack in NDN. In *IEEE ISCC*, 2016.

[177] Stefano Salsano, Andrea Detti, Matteo Cancellieri, Matteo Pomposini, and Nicola Blefari-Melazzi. Transport-layer Issues in Information Centric Networks. In *Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking*, ICN '12, pages 19–24, New York, NY, USA, 2012. ACM.

[178] Jonnahtan Saltarin, Eirina Bourtsoulatze, Nikolaos Thomos, and Torsten Braun. NetCodCCN: A network coding approach for content-centric networks. In *35th Annual IEEE International Conference on Computer Communications, INFOCOM 2016, San Francisco, CA, USA, April 10-14, 2016*, pages 1–9, 2016.

[179] J. Samain, G. Carofiglio, L. Muscariello, M. Papalini, M. Sardara, M. Tortelli, and D. Rossi. Dynamic Adaptive Video Streaming: Towards a Systematic Comparison of ICN and TCP/IP. *IEEE Transactions on Multimedia*, 19(10):2166–2181, Oct 2017.

[180] Klaus Schneider, Cheng Yi, Beichuan Zhang, and Lixia Zhang. A Practical Congestion Control Scheme for Named Data Networking. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, ACM-ICN '16, pages 21–30, New York, NY, USA, 2016. ACM.

[181] Karen Seo and Stephen Kent. Security Architecture for the Internet Protocol. RFC 4301, December 2005.

[182] Ivan Seskar, Kiran Nagaraja, Sam Nelson, and Dipankar Raychaudhuri. MobilityFirst Future Internet Architecture Project. In *Proceedings of the 7th Asian Internet Engineering Conference*, AINTEC '11, pages 1–3, New York, NY, USA, 2011. ACM.

[183] 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)*, pages 1–6, Feb 2015.

[184] J. Shi and B. Zhang. NDNLP: A link protocol for NDN, 2012. https://named-data.net/wp-content/uploads/TRLinkProtocol.pdf.

[185] A. Al Shidhani et al. Reducing re-authentication delays during UMTS-WLAN vertical handovers. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–5, 2008.

[186] A. Al Shidhani and V. C. M. Leung. Local fast re-authentication protocol for 3G-WLAN interworking architecture. In *2007 Wireless Telecommunications Symposium*, pages 1–8, 2007.

[187] C. Sieber, T. Hofeld, T. Zinner, P. Tran-Gia, and C. Timmerer. Implementation and user-centric comparison of a novel adaptation logic for DASH with SVC. In *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, pages 1318–1323, May 2013.

[188] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman. BOLA: Near-optimal bitrate adaptation for online videos. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, April 2016.

[189] Neil Spring et al. Measuring ISP Topologies with Rocketfuel. *IEEE/ACM Trans. Netw.*, 2004.

[190] Volker Stocker, Georgios Smaragdakis, William Lehr, and Steven Bauer. The growing complexity of content delivery networks: Challenges and implications for the Internet ecosystem. *Telecommunications Policy*, 41(10):1003 – 1016, 2017. Celebrating 40 Years of Telecommunications Policy A Retrospective and Prospective View.

[191] E. Muramoto T. Yoneda, R. Ohnishi and J. Burke. Consumerdriven adaptive rate control for real-time video streaming in named data networking, booktitle = Internet Conference 2015, year = 2015, location = Toyoko , JAPAN, pages = 23–32, numpages = 9, publisher = IC2015.

[192] J. S. Thainesh, Ning Wang, and R. Tafazolli. Reduction of core network signalling overhead in cluster based LTE small cell networks. In *2015 IEEE 20th International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD)*, pages 226–230, Sept 2015.

[193] Robert H. Thomas. A Resource Sharing Executive for the ARPANET. In *Proceedings of the June 4-8, 1973, National Computer Conference and Exposition*, AFIPS '73, pages 155–163, New York, NY, USA, 1973. ACM.

[194] Leigh Torgerson, Scott C. Burleigh, Howard Weiss, Adrian J. Hooke, Kevin Fall, Dr. Vinton G. Cerf, Keith Scott, and Robert C. Durst. Delay-Tolerant Networking Architecture. RFC 4838, April 2007.

[195] R. Tourani, S. Misra, T. Mick, and G. Panwar. Security, Privacy, and Access Control in Information-Centric Networking: A Survey. *IEEE Communications Surveys Tutorials*, 20(1):566–600, Firstquarter 2018.

[196] D. Trossen and G. Parisis. Designing and Realizing an Information-centric Internet. *IEEE Communications Magazine*, 50(7):60–67, 2012.

[197] O. Ureten and N. Serinken. Wireless security through RF fingerprinting. *Canadian Journal of Electrical and Computer Engineering*, 32(1):27–33, 2007.

[198] Markus Vahlenkamp and et al. Enabling ICN in IP Networks using SDN. In *IEEE ICNP*, pages 1–2, 2013.

[199] A. Vakali and G. Pallis. Content delivery networks: status and trends. *IEEE Internet Computing*, 7(6):68–74, 2003.

[200] András G Valkó. Cellular IP: a new approach to Internet host mobility. *ACM SIGCOMM CCR*, 29(1):50–65, 1999.

[201] Vassilios G Vassilakis, Bashar A Alohali, ID Moscholios, and Michael D Logothetis. Mitigating distributed denial-of-service attacks in named data networking. In *Proceedings of the 11th Advanced International Conference on Telecommunications (AICT), Brussels, Belgium*, pages 18–23, 2015.

[202] Luca Veltri and et al. Supporting Information Centric Functionality in Software Defined Networks. In *IEEE ICC*, pages 6645–6650, 2012.

[203] Marko Vukolić. The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication. In Jan Camenisch and Doğan Kesdoğan, editors, *Open Problems in Network Security*, pages 112–125, Cham, 2016. Springer International Publishing.

[204] Tao Wan, Evangelos Kranakis, and Paul C. van Oorschot. S-RIP: A Secure Distance Vector Routing Protocol. In Markus Jakobsson, Moti Yung, and Jianying Zhou, editors, *Applied Cryptography and Network Security*, pages 103–119, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[205] Kai Wang, Huachun Zhou, Yajuan Qin, Jia Chen, and Hongke Zhang. Decoupling malicious interests from pending interest table to mitigate interest flooding attacks. In *Globecom Workshops (GC Wkshps), 2013 IEEE*, pages 963–968. IEEE, 2013.

[206] Liang Wang, O. Waltari, and J. Kangasharju. MobiCCN: Mobility support with greedy routing in Content-Centric Networks. In *2013 IEEE Global Communications Conference (GLOBECOM)*, pages 2069–2075, Dec 2013.

[207] G. White and G. Rutz. Content delivery with content-centric networking. https://www.cablelabs.com/wp-content/uploads/2016/02/Content-Delivery-with-Content-Centric-Networking-Feb-2016.pdf.

[208] R White. Securing BGP through secure origin BGP. 6, 01 2003.

[209] White paper: Cisco Visual Networking Index (VNI): Forecast and methodology, 2017–2022. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html.

[210] Dr Gavin Wood. Ethereum: a Secure Decentralised Generalised Transaction Ledger. 2016.

[211] H. Wu and et al. On Incremental Deployment of Named Data Networking in Local Area Networks. In *ACM/IEEE ANCS*, pages 82–94, 2017.

[212] Ziyu Xiao and Harry Perros. Response Time of the S1 and X2 Handover Procedures Between (H)eNBs in a Virtualized Environment. 2015.

[213] Mengjun Xie, Indra Widjaja, and Haining Wangand. Enhancing cache robustness for content-centric networking. In *2012 Proceedings IEEE INFOCOM*, pages 2426–2434, March 2012.

[214] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos. A Survey of Information-Centric Networking Research. *IEEE Communications Surveys Tutorials*, 16(2):1024–1049, Second 2014.

[215] A. Yener and S. Ulukus. Wireless Physical-Layer Security: Lessons Learned From Information Theory. *Proceedings of the IEEE*, 103(10):1814–1825, 2015.

[216] K. Zeng et al. Non-cryptographic authentication and identification in wireless networks [Security and Privacy in Emerging Wireless Networks]. *IEEE Wireless Communications*, 17:56–62, 2010.

[217] Changwang Zhang, Jianping Yin, Zhiping Cai, and Weifeng Chen. RRED: robust RED algorithm to counter low-rate denial-of-service attacks. *IEEE Communications Letters*, 14(5), 2010.

[218] F. Zhang, Y. Zhang, A. Reznik, H. Liu, C. Qian, and C. Xu. A transport protocol for content-centric networking with explicit congestion control. In *2014 23rd International Conference on Computer Communication and Networks (ICCCN)*, pages 1–8, Aug 2014.

[219] Fan Zhang, Wenbo He, Xue Liu, and Patrick G. Bridges. Inferring Users' Online Activities Through Traffic Analysis. In *Proceedings of the Fourth ACM Conference on Wireless Network Security*, WiSec '11, pages 59–70, New York, NY, USA, 2011. ACM.

[220] Guoqiang Zhang, Yang Li, and Tao Lin. Caching in Information Centric Networking: A Survey. *Comput. Netw.*, 57(16):3128–3141, November 2013.

[221] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang, et al. Named data networking. *ACM SIGCOMM Computer Communication Review*, 44(3):66–73, 2014.

[222] Lixia Zhang et al. Named Data Networking. *ACM SIGCOMM CCR*, 44(3):66–73, 2014.

[223] 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)*, pages 83–88, April 2016.

[224] Yu Zhang, Hongli Zhang, and Lixia Zhang. Kite: A mobility support scheme for ndn. In *ACM Conference on Information-Centric Networking*, pages 179–180. ACM, 2014.

[225] Yu Zhang, Hongli Zhang, and Lixia Zhang. Kite: A Mobility Support Scheme for NDN. In *Proceedings of the 1st ACM Conference on Information-Centric Networking*, ACM-ICN '14, pages 179–180, New York, NY, USA, 2014. ACM.

[226] Z. Zhang, Y. Yu, H. Zhang, E. Newberry, S. Mastorakis, Y. Li, A. Afanasyev, and L. Zhang. An Overview of Security Support in Named Data Networking, year=2018. *IEEE Communications Magazine*, 56(11):62–68, November.

[227] Z. Zhu et al. Let's ChronoSync: Decentralized dataset state synchronization in Named Data Networking. In *IEEE International Conference on Network Protocols*, pages 1–10, 2013.

[228] Hubert Zimmermann. OSI Reference Model-The ISO Model of Architecture for Open Systems Interconnection. *IEEE Transactions on Communications*, 28(4):425–432, 1980.

[229] G. Zyskind, O. Nathan, and A. Pentland. Decentralizing Privacy: Using Blockchain to Protect Personal Data. In *2015 IEEE Security and Privacy Workshops (SPW)*, volume 00, pages 180–184, May 2015.