

Routing as a Service Solution for IP-Based Services: An Evolutionary Approach to Introducing ICN in the Real World

Sung-Yeon KIM^{†a)}, Sebastian ROBITZSCH^{††}, Hongfei DU^{††}, and Dirk TROSSEN^{††}, *Nonmembers*

SUMMARY Information-centric networking (ICN) has been positioned for a number of years as a possible replacement to the IP-based Internet architecture with key promises in terms of network efficiency, privacy, security and novel applications. However, such wholesale replacement of the IP-based Internet through a new routing and service infrastructure has always been marred by the difficulties to gain adoption through existing stakeholders and market players, particularly solution providers. In this paper, we provide an evolutionary approach to introducing ICN in the real world by positioning an ICN-based solution as a routing-as-a-service offering for existing IP-based solutions. With this, we enable the expected benefits of ICN for the existing service and application basis of the current Internet. We will outline how we achieve this evolutionary introduction and how existing IP as well as HTTP-based services will be realized. An introduction into our gateway platform will be given, while also outlining first results from a recent showcase deployment.

key words: ICN, NGN architecture, surrogacy

1. Introduction

The consumption of media-rich content, such as video, has been driving the capacity build-out of major ISPs in recent years. Forecasts expect that particularly personalized viewing will account for more than 80% of traffic by 2019 [10]. This forecast creates an economic pinching point for many operators since efficient (IP) multicast cannot be used for these type of services. As a consequence, costs for deployments as well as services are continuously pushed up since video is delivered as individual content to every user, creating a linear dependency between cost and number of users. At the same time, Quality of Experience at the user level is often negatively impacted due to the capacity impasse that is often experienced at peak times.

In order to counter this issue, content delivery networks (CDNs) are deployed that place popular content closer to end users through peering arrangements between content and network providers. Such CDNs effectively create a service-level routing solution that reroutes requests to a nearby replica. However, this routing solution is realized through rather inefficient DNS-level indirections and only allowing for longer-term changes in these indirections towards new, possibly better located replicas due to changed constraints (e.g., new demand patterns, changes in network

load). What is missing is a true network-level service routing solution that would allow for multicast delivery of same content pieces to end users, removing the pressure for personalized viewing scenarios, while also enabling rapid changes for alternative layout points for resilience and load balance purposes.

It is here where information-centric networking (ICN) comes into play. Positioned at the network level in its original proposition [12], [13], ICN effectively acts as a routing solution for a different, pull or pub/sub based, service interface. Existing ICN works, however, require changes to service interfaces and therefore ultimately applications, which constraints the adoption at the application level. On the other hand, ICN often comes with significant changes to network infrastructure, creating reluctance at the operator level to adopt the new architecture.

With our solution in this paper, we take a different approach to introducing ICN in the real world.

Instead of a wholesale replacement of IP-based network infrastructures, we position our ICN platform as a *Routing-as-a-Service* (RaaS) that sits as a shim layer in-between the IP/HTTP service interfaces of IP endpoints (servers and clients) and the transport infrastructure of an operator. It is positioned as a RaaS offering due to the pure software basis for its realization, allowing for a service-oriented deployment in a slice-like manner in SDN-based environments as well as future 5G environments. Through this approach, we deliver on the benefits of ICN in terms of increased network utilization, while preserving the rich and existing ecosystem of IP-based Internet services.

Our platform is realized by marrying ICN concepts with increasingly deployed capabilities of software-controlled Layer 2 networks (enabled through SDN), leading to a solution that reduces operational costs for network operators by allowing multicast delivery of HTTP unicast responses for the same content. With this, our RaaS solution solves a concrete problem, namely an operational cost reduction that stems today's linear cost increase for HTTP-based services, particularly video ones, without the need for any IP endpoint changes. Our solution also reduces service-level latency through allowing for the flexible placement as well as quick activation of surrogate servers closer to the end user.

The remainder of the paper is organized as follows. In Sect. 2, we provide a brief overview of the ICN flavour that is at the heart of the proposed solution, followed by our solution design in Sect. 3, which outlines the main components of our RaaS solution. The design of our implemented platform is

Manuscript received June 8, 2016.

Manuscript revised July 20, 2016.

[†]The author is with InterDigital Asia, LLC., Seoul, Republic of Korea.

^{††}The authors are with InterDigital Europe, Ltd., London EC2A 3QR, United Kingdom.

a) E-mail: sung-yeon.kim@interdigital.com

DOI: 10.1587/transcom.2016CNI0002

presented in Sect. 4, while we outline the opportunity space for our solution in Sect. 5. We then provide an overview of a showcase deployment in Sect. 6, before concluding the paper in Sect. 7.

2. RaaS and ICN: A Match Made in Heaven?

In this section, we will discuss why we see ICN and the problem of improving on routing in the existing Internet are a perfect match. In order to understand this viewpoint, some brief background into the utilized ICN flavour is required before laying out our thinking on routing as a service.

2.1 ICN Background

The area of ICN comes with a number of ‘flavours’ that realize the main concepts through different solutions. While specifically basing our solution on the work in [6], we attempt in the following to provide a more general introduction into the concepts we are utilizing.

Intuitively, any ICN architecture places *information* at its heart. Hence, we assume some means to exist for identifying individual information items. Such identification can be realized, for instance, through statistically unique fixed-size labels as used in [6]. Individual information items can be called into a context; called *scoping*. Each information item is placed in at least one scope. With scopes being sets of information, they are information themselves. Hence, they are identified through the same naming mechanism and can be nested in another scope, building directed acyclic graphs (DAGs) of information over which computation can take place [6]. Namespaces can exist in parallel due to the support for arbitrary root scopes, assuming some form of identifier avoidance procedure, e.g. in the form of standardization for such root scopes.

The architecture in [6] exposes a specific, here publish/subscribe, service model to applications. Hence, information items are published by those who are willing to offer them, and subscribed to by those who are interested in them, providing spatial and temporal decoupling of the communication. Other service models, such as the content pull model on [11], can easily be realized through such pub/sub model.

The service semantics are realized through three core functions, namely *rendezvous*, *topology management* and *forwarding*. The rendezvous (RV) function matches the demand and supply of information, thus creating a pub/sub relationship at any point in time. As the result of such matching, the RV function initiates the creation of a communication relation between the matched publisher(s) and subscriber(s) via the topology management (TM) function. After receiving such request, the TM function selects the appropriate network resources according to the optimization objective of the underlying routing algorithm.

Various forwarding mechanisms, such as line-speed publish/subscribe internetworking (LIPSIN) [7], can be used to represent a suitable communication relationship. Here, each directed edge in the network is assigned a fixed-size

Bloom Filter (BF)-based Link Identifier (LId). A path in the network then represents all those LIDs along the path from the source to the sink and the forwarding decision is realized through a simple binary OR operation over all LIDs along the chosen path. This simply check is realized by the forwarding (FN) function as bitwise AND and COMPARE logical operations. With this, multicast delivery of packets becomes a natively supported operation.

The separation of TM and FN function well aligns with the emergence of SDN-based transport networks. The path computation, performed in the TM function, for LIPSIN-based forwarding function determines forwarding information that any OpenFlow-compliant switch can easily act upon with simple matching rules, see [15], while the TM function interacts with a standard SDN controller for the setup of those matching rules. All of this is achieved with a purely proactive rule insertion and no need for flow-based state in the SDN switches.

2.2 ICN and the Internet Routing Problem

Looking carefully at the main concepts of ICN, we come to realize that providing IP-based services is inherently an ICN problem, as the communication between two endpoints at IP level is inherently an information exchange problem, an aspect becoming even more apparent at HTTP level. However, the introduction of ICN has faced significant obstacles, represented in lengthy standardization and implementation efforts, since replacing the IP communication stack of existing and future IP endpoints with an ICN compliant one is a challenging and rather long-term task. While one could argue that there is no necessity to change anything in the way packets are routed through the internet, we cannot see how the firm requirements for 5G (specifically extremely low latencies and higher throughput) can be achieved without rethinking how routing is realized, particularly when attempting to provide such environment in a backward-compatible manner from an application perspective by preserving IP and HTTP service abstractions at the endpoints.

It is here where our proposed RaaS solution comes into play by clearly separating the routing and forwarding functions and utilising ICN semantics [6], [12] to allow on-the-fly integration of new service endpoints at the IP level. With this we achieve routing as a service by virtue of a softwarized path computation element, with the forwarding being realized over a highly efficient (e.g. SDN-based) forwarding fabric [15]. The software-based routing is realized in ICN components that architecturally represent the rendezvous and topology management functions of existing ICN solutions [6], where such functions can implement a variety of constrain-based routing algorithms, each of which can easily be exchanged in a plug-and-play service manner to accommodate various 5G service requirements.

What makes such RaaS viewpoint doable in operational networks, removing them from purely lab-based test environments and opening the possibility of deployments at operator scale, is the emergence of software-controlled networking

(SDN) concepts as well as solutions. With many operators looking towards SDN equipment as a viable alternative to special-made transport infrastructure, we argue that our ICN basis as described in Sect. 2.1 perfectly matches with the emerging deployment of SDN infrastructure, as the separation of routing (i.e. path computation) and forwarding (i.e. acting on this path information) perfectly aligns with the fundamental control/data plane separation in SDN. This is showcased in Sects. 3 and 4, with a solution and platform design that well integrates with SDN platforms that are currently deployed at operator scale, while complementing the solution with a protocol translation from IP/HTTP to ICN in edge-based gateways that interface with the SDN-based transport infrastructure, as described in more detail in the following section.

3. Solution Design

In this section, we provide an overview of the system architecture and our solutions for mapping IP as well as HTTP transactions onto ICN operations.

3.1 Starting Points

In the attempt to rethink the current Internet architecture, a clean-slate approach has created a trajectory that positions ICN as a global replacement for an IP-based Internet as we know it [12]. However, such a trajectory comes with drawbacks, many of which the ICN community has suffered from in recent years. For instance, the replacement of IP as the main internetworking protocol not only comes with the burden of heavy standardization, but also requires agreement among many stakeholders in the current Internet, ranging from operators and vendors over software developers and end-device makers to policymakers. Furthermore, the interconnected nature of the Internet due to its Autonomous Systems (AS) requires viable methods for truly scalable internetworking of individual ICN deployment islands, with solutions to this important problem still being in their infancy. While early motivations for ICN [11] included the increasingly content-centric manner of consuming current Internet services, novel Internet services, such as those made possible by the Internet-of-Things (IoT) were also proposed as a possible driver for introducing a new inter-networking layer. However, many of these promises of new services have not yet materialized.

In this paper, we consider a different trajectory for ICN introduction into the market. We propose to harness the innovation potential of IP-based applications and solutions, while benefitting from specific ICN solutions in terms of their potential for better performance compared to their IP-based counterparts. With this in mind, we assume no changes to existing endpoints, while providing a purely software-based solution of edge-based protocol mapping with a SDN-compliant routing solution in the transport network. The next section will outline our system approach to this starting point.

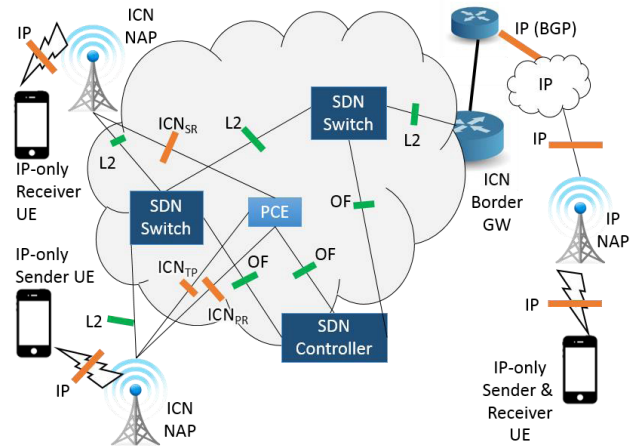


Fig. 1 System architecture of the proposed RaaS solution.

3.2 System Architecture

Figure 1 outlines the main components of our architecture. The path computation entity (PCE) performs the joined RV and TM function of our presumed ICN solution, while standard SDN switches are used for the FN function. The PCE interfaces with the SDN controller through standard OpenFlow (OF) interfaces, shown in green [15].

In order to preserve the IP interfaces towards user equipment (UE), our ICN architecture uses a gateway approach, represented as the Network Attachment Points (NAPs), e.g., the access gateways from customers to the network, and the ICN border gateway (ICN BGW), i.e. the access from and to peering networks or operator server resources in Fig. 1. These gateways handle all the offered protocols at the IP interface, e.g., IP, TCP, HTTP, COAP. Furthermore, the NAP provides standard gateway functions such as network address translation (NAT), firewalling and dynamic IP address assignment, wherever it is required. With this, the network appears like a standard IP network to UEs as well as peering networks, while the NAPs and GW interface through ICN-based interfaces with the PCE (shown in orange) for the desired routing functionality, as explained in the following subsections for the IP as well as HTTP protocol abstractions.

3.3 IP-over-ICN Protocol Mapping

In this subsection, we provide the design of specific namespace and the operations in IP-over-ICN utilizing this namespace for protocol mapping. In general, many operator's IP network configurations use prefix-based addressing, i.e. the provided IP range of an operator is provided through a prefix range while peering operations are defined through such prefixes, too. The prefix-based addressing provides a natural form of address space aggregation which is already used in Internet and which we intend to support. Note that the prefix-based addressing allows for the operator to easily assign sub-spaces to customers, e.g. a data centre client

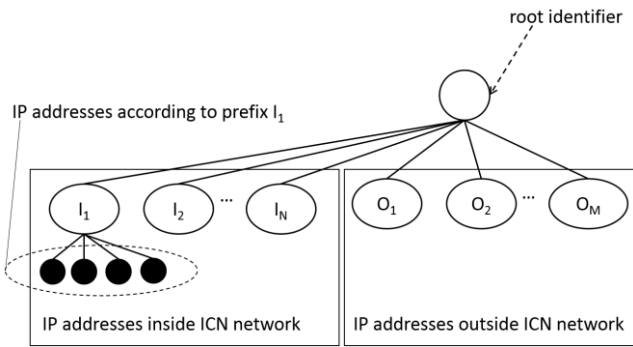


Fig. 2 The namespace for prefix-based addressing.

can be assigned a prefix within the operator’s address space, providing a large number of individual IP addresses to this client.

With this in mind, the mapping of IP-based communication to ICN interprets IP packets as being content that is sent to a particular destination, i.e. the destination becomes the content name (i.e. the source IP address), reflected in a suitable ICN namespace. For the design of such ICN namespace to handle the IP protocol translation, we envision that a separate scope identifier, even a *root identifier* is chosen for IP-like communication. A root identifier would allow for separating IP-like communication from other ICN communication, e.g. for operational or migration reasons. Hence, the root identifier could be standardized or simply defined in a proprietary manner across ICNs of vendors or operators. Under this root scope, there exists so-called *prefix scopes*, both for operator-internal IP prefix ranges as well as for external IP prefix ranges, shown as I_1 to I_N and O_1 to O_M , as depicted in Fig. 2.

These prefix scopes are determined through a combination of the IP address and subnet mask information of IP routing prefixes. In other words, the IP address and subnet mask information is hashed into a single identifier of fixed length to determine a statistically unique scope identifier that represents the specific IP prefix information. For IP addresses outside the ICN network, we envision that these prefixes are calculated and published under the root scope by an operator management system, based on known IP prefix configurations of the IP border GWs of the operator. In our initial implementation, these prefixes are included in a configuration file for each NAP and GW, distributed at start-up time.

In Fig. 2, we indicate black circles as the actual information identifiers for the IP addresses according to the IP prefix information that is represented by its father scope, i.e. I_1 . The information identifier is formed by hashing a fully qualified IP address. Thereafter, each scope contains information items and their data that is compliant to the IP prefix information that the scope represents.

3.4 HTTP-over-ICN Protocol Mapping

To realize communication to any HTTP server device, the

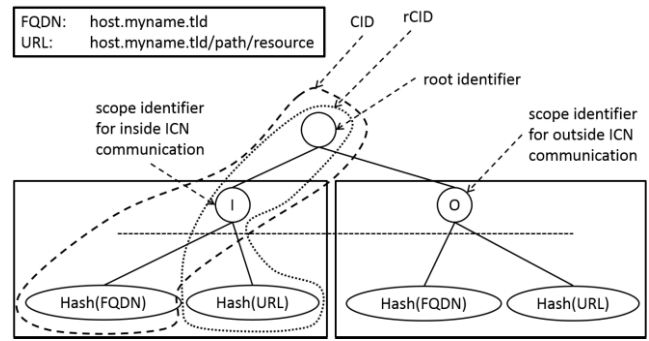


Fig. 3 The namespace for HTTP-based services.

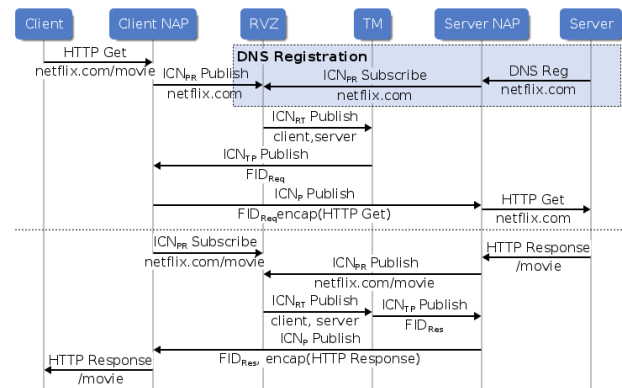


Fig. 4 Message sequence chart for HTTP-over-ICN.

NAP and GW utilizes a namespace as depicted in Fig. 3. The Domain Name System (DNS) namespace within the ISP is grouped under a single scope (illustrated with I) while all other addresses are grouped under the sub-scope O. A communication (HTTP request) with an HTTP server is a publication to its fully qualified domain name (FQDN), represented as *Content Identifier (CID)* in Fig. 3, while the HTTP responses are delivered to the URL, hashed over a carefully selected set of request parameters, of which the URL is only one. This response is represented as *reverse CID (rCID)* in Fig. 3. An HTTP request from a client of any device is interpreted as an appropriate ICN name (CID in Fig. 3) which can be related to the FQDN of the server, while the response to that request is published to the appropriate ICN name related to the URL of the request. This allows a server NAP to simply subscribe to the FQDN of any attached web server, while the web server can publish any response to the corresponding URL.

In Fig. 4, we show an example of message sequence chart (MSC) for a client HTTP request and server HTTP response. We have labelled cNAP and sNAP as the NAP at the client and that at the server side, respectively.

In the figure, we first show that the server registers its DNS name (netflix.com here), which leads to its NAP subscribing to the server’s FQDN. Then, an HTTP request for netflix.com/movie is received by the client’s NAP, which publishes it to the Rendezvous (RVZ) (as part of the PCE);

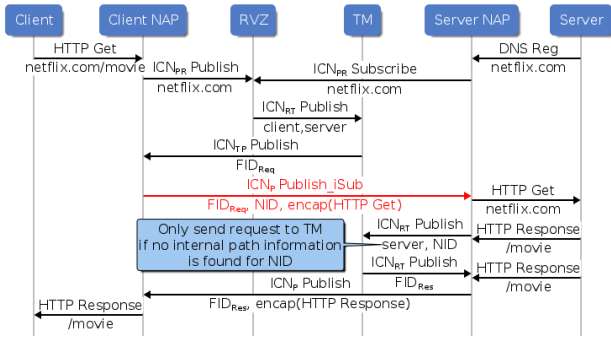


Fig. 5 Reducing latency by RVZ interaction at the sNAP.

the RVZ matches it with the server’s subscription and asks the TM (as part of its internal realization) to create a forwarding path for the request, which is eventually sent to the client’s NAP, which finally sends the request to the server’s NAP, where it is de-capsulated and passed as a new HTTP request to the server. The client’s NAP then subscribes to the full URL of the request, so that when the HTTP response arrives from the server and the server’s NAP publishes it, the RVZ will match it, ask the TM for a path, and the server’s NAP will receive the path, so that it may send the response to the client’s NAP, where it will be de-capsulated and sent as a regular HTTP response to the client. It is clear that the full messaging in Fig. 4 cannot yield in high performance due to the required RV/TM interactions for every request and response interaction. The first improvement is that of realizing the rendezvous function for the response publication in the sNAP itself, therefore removing the need to initiate the sNAP-RVZ interaction. For this, we extend the ICN_F interface in Fig. 1 for the HTTP-over-ICN mapping to include the Node Identifier (NID) of the cNAP in the publication message. With this, the sNAP can create a relation between the incoming publication and the response publication without needing to contact the RVZ. Furthermore, this NID can be used at the sNAP to initiate the path computation via the TM by utilizing its own NID and the NID of the client. Figure 5 shows the resulting message sequence indicating in red the changed message, represented as a new ICN primitive called *publish_iSub()* with the inclusion of the NID constituting an implicit subscription to the response when publishing the request.

However, Fig. 5 still shows one cause for latency that we would like to remove, namely the path computation initiated by the sNAP. When requesting a path from the client to the server, it only constitutes a small overhead to the TM to not only compute the forward path, i.e. from cNAP to sNAP, but also the reverse path, by simply taking the LIDs of the reverse directional links on the path and combining them to a second BF identifier, namely that of the return path, i.e. from sNAP to cNAP. Hence, we propose to extend the signalling on the ICN_{TM} interface to include an option for a return path inclusion in the response. Furthermore, we propose to include this return path identifier in the above introduced *publish_iSub()* message. The result of these changes is shown

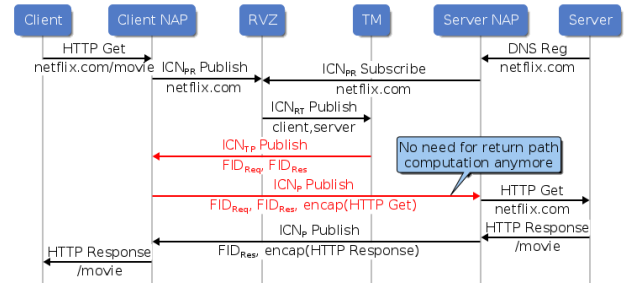


Fig. 6 Reducing the latency due to TM interaction at the sNAP.

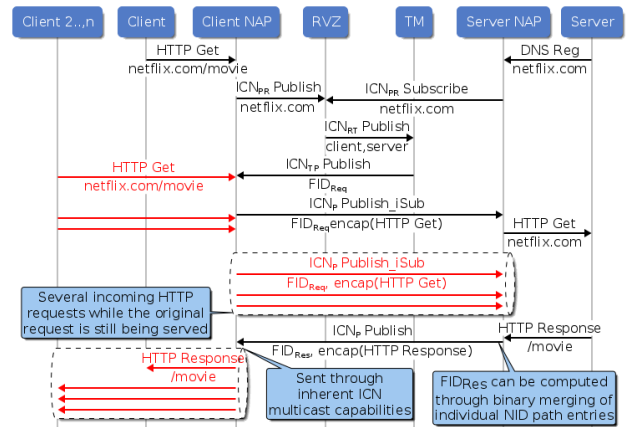


Fig. 7 Spontaneous multicast group formation through BF merging.

in Fig. 6.

However, as stated before, the inclusion of the NID can further improve latency if we amend the state management in the NAP by adding a mapping from NID to Forwarding Identifier (FID) to the already existing mapping from RID to FID. In other words, in addition to supporting the repeated communication within a given information identifier, i.e. supporting channel semantics and as being used in the forward path of the HTTP request, we suggest to maintain a table for repeated communication to particular nodes, the latter identified through their NIDs.

A significant advantage of the proposed amended state management at the sNAP is that of spontaneous multicast group formation, depicted in Fig. 7.

In order to efficiently implement the co-incidental multicast capability, i.e. the ability of sending responses to the same HTTP requests to a group of requesting clients, it is crucial that we are able to spontaneously create multicast groups that are formed of the clients that have quasi-synchronously requested the same response. It is clear to see that creating TM requests for such specific responses is undesirable, due to the additional delay caused by the sNAP-TM interaction. Instead, we can utilize an interesting characteristic of the underlying BF-based forwarding solution. BF identifiers that describe a path in the network, i.e. from the sNAP to a specific cNAP, can be used to create joined paths by simply OR-calculation of the individual BF identifiers into a new one. Hence, when the sNAP receives multiple HTTP

requests for the same resource from other cNAPs while waiting for the first HTTP response from the server, it forms a co-incidental multicast group with all cNAPs awaiting the HTTP response. Therefore, the amended state management will allow to simply look up all FIDs to the NIDs that have outstanding subscriptions, OR-calculation of all FIDs into a new BF and use this BF in the response, which is now delivered to all the nodes of the co-incidental response.

4. Platform Design

In this section, we consider a network attachment point (NAP) architecture, NAP deployment scenarios, and the software defined network (SDN) integration on the NAP.

4.1 Network Attachment Point

The overall NAP architecture and some function blocks are depicted in Fig. 8. Note that all boxes drawn in a dotted line represent conventional IP and existing ICN Blackadder functionality which is not part of the NAP design. All boxes drawn with a solid line are related functional components which will be briefly described in the following subsections.

The starting point for all IP-based communication is an IP endpoint which uses a standard IP stack to send and receive packets. The IP endpoint communicates with an IP Gateway.

The communication between IP endpoints over the Internet depends on how the endpoints are connected to the Internet. Most customers signed a contract with an ISP which provides a home router taking over the tasks required to connect them to the operator's core network. The home router itself arrives preconfigured at the customer's premises so that it can automatically connect to the operator's core network, where all traffic breaks out into the Internet.

The provided home router as a standard IP Gateway performs standard gateway functionality such as IP assignment to the IP endpoints, e.g., DHCP, firewalling, NAT. As the aforementioned functionality can be found in all routers provided by rental line operators, the IP GW is considered to be fully operational when connecting IP endpoints to an information-centric architecture through a NAP. The ICN core node platform is the interface towards the information-centric network. As the underlying ICN core node platform,

the NAP will utilize the PURSUIT prototype Blackadder [6]. This service model allows for publishing and subscribing operations over an information model that is based on a directed acyclic graph, following the PURSUIT architecture [14].

Blackadder uses the Click software router platform which allows it to run either on user or kernel space on Linux, Mac OS X and FreeBSD machines, as well as on the ns-3 network simulator. In relation to this ICN core node platform, the NAP acts as a publisher/subscriber application, utilizing the publicly exported application interfaces of the platform. The underlying Layer 2 network represents the forwarding layer which is utilized to carry the ICN packets between two network nodes.

The de-multiplexer (demux) captures IP carrying Ethernet packets from the IP Gateway or the publicly reachable endpoints connected to the NAP. The demux uses Layer 2, 3 and 4 information to determine the handler that is most appropriate to handle an incoming IP packet. On Layer 2, the demux detects which Layer 3 protocol is encapsulated.

The same applies to Layer 3 and Layer 4. This not only allows the demux to determine the appropriate handler, but it also allows future abstractions to be easily realized within the NAP implementation.

All IP stacks require the MAC address of the next hop NIC in order to allow the receiving NIC to determine if a particular packet should be kept and processed. For Ethernet frames, this MAC address is used for the link-local Layer 2 communication between the NAP and an IP endpoint served by the NAP. All traffic that uses IP as the network layer protocol, but not HTTP on the application layer, is de-multiplexed into the IP handler following the IP-over-ICN abstraction. The IP handler uses the NAP's internal database to determine whether or not an IP packet can be published in the ICN network; if there is no subscriber available for the information item under which the packet shall be published, the IP handler adds the packet to the NAP's internal IP packet buffer. Furthermore, the IP handler uses state information in the NAP's internal database to check if the scope path and information item has been already published to the ICN network. Whenever this is not the case, the IP handler carries out the required actions to create the desired scope tree branch. When the scope tree has been published, the IP handler advertises the availability of new data under its corresponding scope.

The HTTP handler receives all IP traffic sent via UDP or TCP on transport layer port 80, by the protocol and port demux. Due to the nature of HTTP scenarios, where a client requests content from a web server, there are two main message types specified in [5]: the HTTP request and the HTTP response. As a result, each message has a list of possible methods indicating the type of request or response. The HTTP handler is responsible to determine what method the HTTP message comprises and acts upon this information. If the HTTP message sent by an IP endpoint is identified as a request with method GET or HEAD, the HTTP handler uses the HTTP-over-ICN abstraction to publish this information

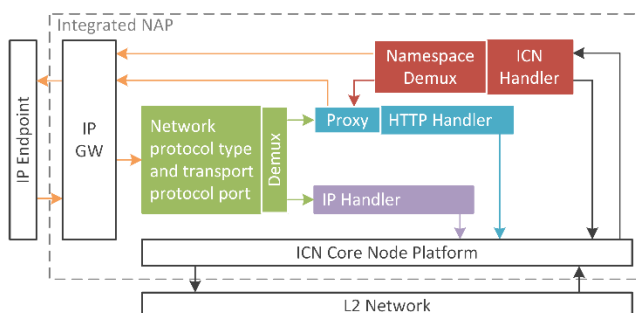


Fig. 8 NAP architecture.

in case it cannot be served by the handler’s web proxy.

The proxy located within the HTTP handler performs similar tasks as proxies deployed in the current Internet, i.e. transient caching and serving IP endpoints with cached DNS entries. Furthermore, the TCP session towards an IP endpoint served by the NAP is terminated in the proxy, as the HTTP-over-ICN abstraction only uses HTTP message header information to publish information. As every HTTP request message traverses the proxy, any content which has been cached prior to this request is provided immediately to the IP endpoint and will not be published again into the ICN network. The same logic applies to ICN packets received from remote NAPs. Any HTTP request de-capsulated from an incoming ICN packet is processed by the proxy through the HTTP handler. In case the requested content element is available in the proxy’s cache, the request is not sent to the IP endpoint but immediately pushed towards the MIME demux as a plain HTTP response packet. If an incoming ICN packet encapsulates an HTTP response it also traverses the proxy before being sent towards the IP endpoint that requested this content element. This allows the proxy to cache the content element in case it is requested again by any of the IP endpoints served by the NAP that received the HTTP response.

The demux depicted allows the de-capsulation of ICN packets targeted at IP endpoints. Using the root scope of the ICN ID the appropriate namespace is determined identifying whether the ICN payload is an IP packet or an HTTP message. In case it is an IP packet, it is immediately sent to the IP gateway. If it appears to be an HTTP packet, it is handed over to the HTTP handler.

4.2 NAP Deployment Scenarios

While Sect. 4.1 outlines the general NAP implementation design, it is also important to consider the deployment scenarios for the platform. We describe here two such scenario. First, we consider the deployment at the customer’s premises where the operators’ rental line is used to access the Internet. Second, the NAP can be deployed in aggregation points, such as in the Broadband Remote Access Server (BRAS) element in a fixed ISP network, or in data centre NAPs where aggregation points use routing prefixes to forward traffic to the server with an IP address covered by the routing prefix.

Note that, from the ICN perspective, the ICN BGW that breaks out to the Internet for services not residing inside the ICN network is also a NAP. As will be explained below in further detail, the NAP translates packets sent over an IP stack into POINT’s ICN semantic and vice versa. Therefore, an ICN BGW is nothing more than a NAP serving a particular range of services.

4.2.1 Host-Based Deployments

In Fig. 9, we provide the customer’s premises scenario. In the scenario, a home router which is usually provided by the rental line operator operates a network with private IP

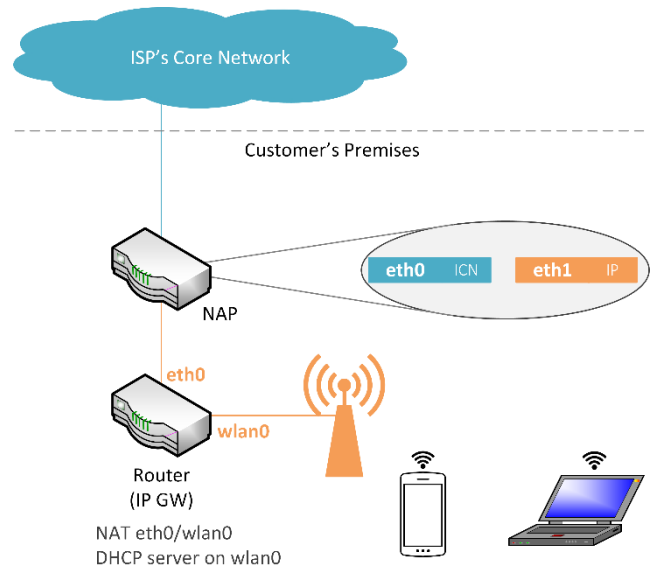


Fig. 9 Host-based NAP deployment scenario.

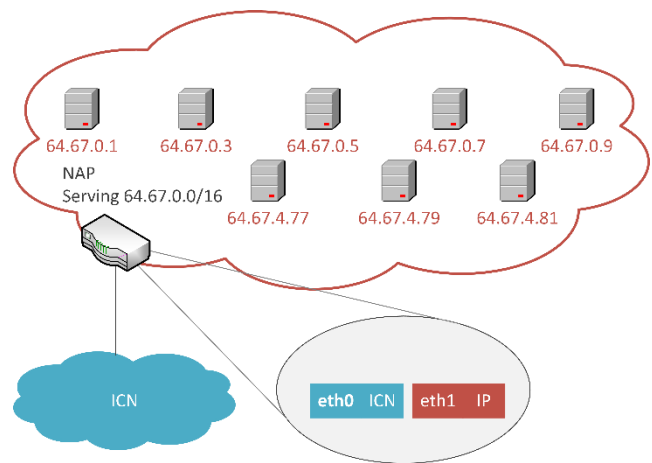


Fig. 10 Routing prefix-based NAP deployment scenario.

addresses where the IP address assignment is done automatically, e.g. via DHCP. The home router then translates all IP traffic so that it appears to originate from the router’s public IP address, provided by the rental line operator, e.g. via PPPoE. Furthermore, the home router also has strict firewall rules which do not allow any unauthorized incoming traffic to be forwarded to one of the IP endpoints.

However, customers can change these firewall (forwarding) rules using a web interface on the router itself. The NAP is placed between the operator’s network and the home router and translates all IP traffic originated and targeted from/to the public customer’s IP address.

4.2.2 Routing Prefix-Based Deployments

In Fig. 10, we provide a scenario in which routing prefixes are used by service providers and content operators to connect a vast amount of IP endpoints to another network without the

need to add all IP addresses to the routing table in the gateway which serves the operator's IP endpoints. The NAP at the edge of the IP endpoints translates all IP traffics originated and targeted from/to the endpoints. Compared to the host-based scenario above, the routing prefix approach requires the NAP to serve multiple IP addresses based on the prefix.

5. Opportunities Space

In today's networks video traffic accounts for more than 50% of the overall traffic, forecasted to reach 80% by 2020. Reason being the penetration of technologies such as 4k and 360° video, virtual/augmented reality and more personalised online streaming scenarios. With most services being delivered via HTTP-level streaming (HLS) the bandwidth cost increases linearly with the number of users. The following three sections describe the opportunity space with the proposed RaaS architecture to bring down the cost for operators to provide a truly 5G-ready network infrastructure.

5.1 Multicast

Video content provided by Video on Demand providers such as YouTube, Netflix, HBO or Amazon is delivered in HTTP unicast sessions in order to ensure a personalised service experience using widely deployed viewing ecosystems, i.e. web-browsers and smartphone/tablet apps. In order to cope with the increasing proliferation of unicast HTTP video the proposed RaaS network architecture with its novel HTTP to ICN translation mechanisms and extensions described in Sect. 3 allows the re-introduction of multicast for HLS scenarios, leading to a first order reduction of the bandwidth costs.

5.2 Surrogacy

In order to enable reasonably low end-to-end delays across the Internet content providers heavily leveraging CDNs which represent an authoritative copy of the content so that a particular web resource is located reasonably close to an end user. To make CDNs transparent to end users DNS redirect (e.g. google.com is resolved into google.co.uk in the UK) and location specific DNS entry (youtube.com results in different IP addresses depending on the user's point of attachment) solutions have been implemented today's networks. However, in order to enable the 5G KPIs extreme low latencies, when needed, and extremely high throughput, the proposed RaaS system allows the placement of authoritative copies of any IP service anywhere in the network at run-time which cannot be achieved using CDNs. This technology — referred to as surrogacy — enables an improved service provisioning resulting in a much shorter path between the IP endpoints which results in a significantly lower latency especially when placing the surrogate environment close to the user.

5.3 Resiliency

One of the key advantages of the proposed ICN-based RaaS

network architecture is that it has been designed with resiliency in mind. As the forwarding of packets does not require any states in the ICN forwarders/routers and routing is fully decoupled from the service provisioning, resiliency is provided natively by the proposed RaaS solution. With the ability of enabling/disabling surrogate servers connected through an sNAP at run-time traffic engineering exercises to accommodate for link-failures, load-sharing and a change in network conditions or end user requirements transparently to serving and requesting end-points is provided by the underlying ICN semantic of the RaaS architecture.

6. Showcase Deployment

Based on our architecture and protocol mapping solutions presented in the previous sections, we have realized our RaaS solution in a showcase deployment at moderate scale with tens of operational network components. Although such deployment does not constitute a pre-commercial deployment, it provides insights into architecture benefits and platform-level performance that we will share in the following sections.

6.1 Use for Showcase Deployment

As stated in Sect. 5, the increasing costs of unicast delivery of video over HTTP are putting an enormous strain on operators and we expect that our RaaS platform provides a concrete solution to lower these costs. Hence, the main objective of our showcase and the preliminary evaluation shown in the following section is to outline expected network utilization benefits. For this, we assume a simple HTTP-level streaming (HLS) scenario in which a number of user happen to watch the same video content at roughly the same time. The word 'roughly' here refers to a statistical coincidence of HTTP chunk retrieval requests within a certain time interval. Unlike IP multicast based IPTV offerings, such synchronicity is not required for the system to work. Our solution merely utilizes such synchronicity in a dynamic per-chunk request manner. The following subsections outline the topology used for our experiment as well the initial results for such use case.

6.2 Topology

Figure 12 illustrates the topology of the test-bed to demonstrate the benefits of the proposed RaaS solution. All IP endpoints (both clients and servers) are depicted with grey squares and their NAPs with aqua circles; each NAP serves exactly one IP endpoint. The Mininet platform [13] was used in this topology to construct a cluster of 10 IP endpoints acting as clients requesting content from a server which located at the centre bottom in Fig. 12. This IP endpoint is labelled as "Apache Server" and serves *http://video.point* to the Mininet clients. Another IP endpoint and its NAP is depicted in the bottom left, next to the RV/TM, which acts as a trigger client to start the experiment. All yellow circles are pure ICN forwarding nodes with no other functionality than connecting

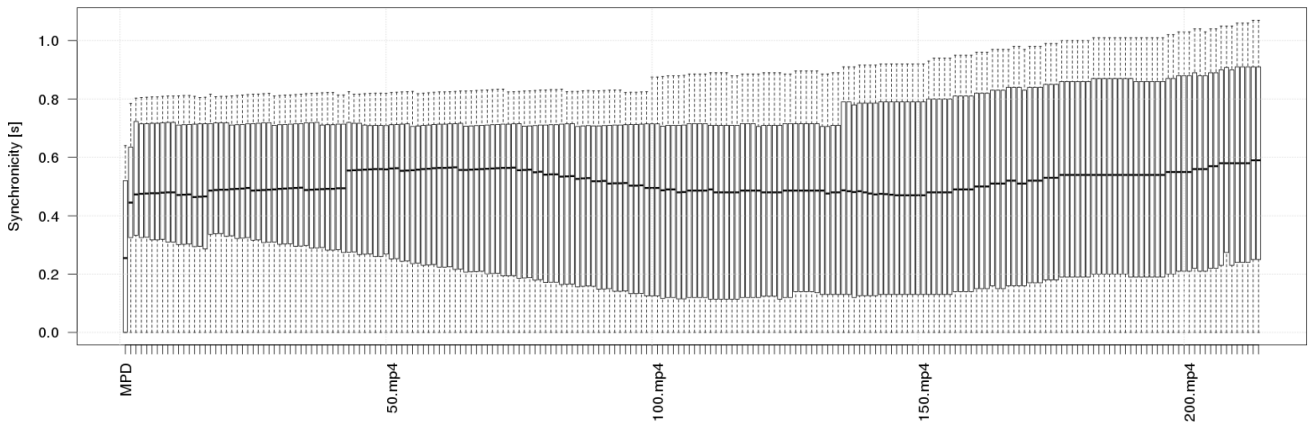


Fig. 11 Synchronicity across all UEs chunk.

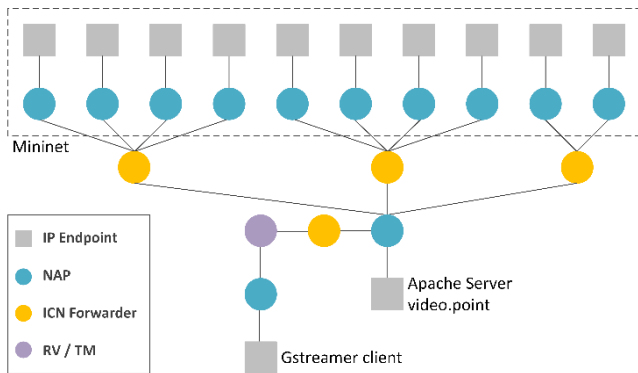


Fig. 12 Test-bed topology with 10 emulated clients.

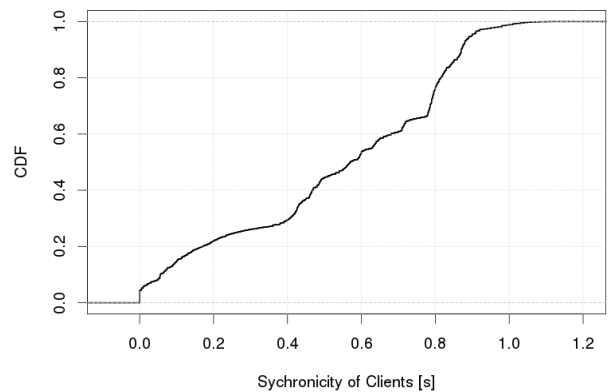


Fig. 13 CDF of the synchronicity of 10 UEs across all HTTP requests.

the neighbouring nodes.

The content offered to all clients by the server is an MPEG DASH encoded video which allows to stream a video via HTTP. To start the experiment the Gstreamer client issues the initial HTTP request to `http://video.point/stream.mpd` which triggers a dedicated software, running on the NAP serving `video.point`, to send an ‘out-of-band’ control message to all emulated clients notifying them to start requesting the `stream.mpd` file too. This mimics a group of clients that happen to watch the same content at roughly the same time.

Version 1.4.4 was used together with an unmodified Apache web server. The MPEG DASH video was encoded with publicly available encoding and packaging tools.

6.3 Synchronicity of Clients

Before presenting the coincidental multicast results of the showcase deployment this section is dedicated to shed some light on the synchronicity of the 10 emulated clients which essentially determines the multicast gain that can be achieved. In order to measure the synchronicity of all UEs across all their HTTP requests a customized software was placed on the NAP serving `video.point` which sniffed for HTTP requests traversing the IP interface towards the IP endpoints and logs them including a timestamp.

To obtain the CDF depicted in Fig. 13 the following

methodology was applied: the timestamp of the first occurrence of a particular HTTP chunk was used as an offset value any following HTTP request asking for the exact same resource. Thus, the resulting list of timestamps (per HTTP resource) are all normalized to 0s and when treating them as a single data set show how synchronized all UEs were across all HTTP requests.

As can be seen in Fig. 13, the mean is around 0.55s with a rather steady slope towards 100% which is caused by the insufficient number of repetitions of the experiment (42 repetitions). The same applies for the CDF towards 0%. The synchronicity per HTTP chunk is illustrated in Fig. 11, allowing insights into the synchronicity over time and if clients tend to become more de-synchronised over time. For each chunk a boxplot is illustrated including the median and the upper and lower quartiles. The very first HTTP request is for the MPD file which tells the DASH client which chunks are available, followed by the `init.mp4` and then sequentially numerated files, displayed on the x-axis.

As can be seen, there is a significant difference in the synchronicity median between the MPD, `init.mp4` and `1.mp4`, compared to the medians of the remaining boxplots. This can be explained by the fact that once the customized trigger software, placed on the NAP serving `video.point`, receives the HTTP request from the Gstreamer client for the

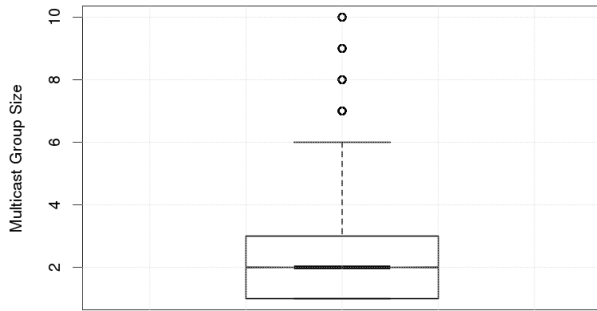


Fig. 14 Boxplot of mean co-incident multicast gain for 10 emulated UEs.

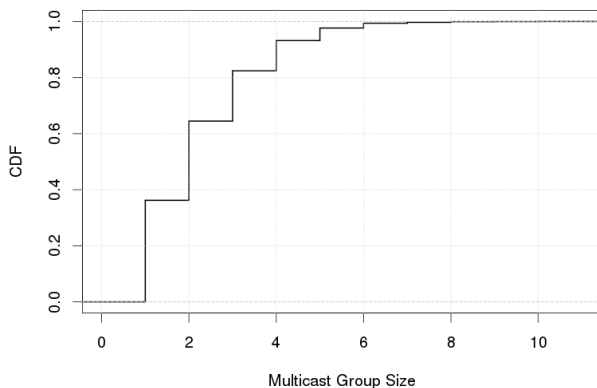


Fig. 15 CDF of co-incident multicast gain for 10 emulated clients.

MPD file all 10 emulated clients receive the trigger to start playing the video too. As the service provisioning results in multiple multicast groups, the subsequent HTTP request from each client are more spread in time than the initial MPD request. Right after the third HTTP request is served (Fig. 11, 2.mpd) the synchronicity across all HTTP chunks keeps constant around 0.5s with a steady increase towards the end of the video indicating that clients become less synchronised over time.

6.4 Experiments

The experimentally obtained multicast group size results over all iterations are presented in Fig. 14 and Fig. 15 using a box-and-whisker and a CDF plot, respectively. The boxplot reveals a very solid median at 2 (notches were enabled) with a lower and upper quartile between 1 and 3 and a maximum at 6. Outliers at 7 through 10 complement the plot. The very central position of the median also indicates that it is not biased towards either quartile. Finally, the CDF of all recorded multicast gains for 10 emulated clients shows a central estimate of 2 (50%) with a large span of approximately 38%–62%. According to the CDF plot, the multicast group size range can be projected to be between 1 (10%) and 4 (90%) which fits well with the boxplot in Fig. 14 and the exemplary multicast over time graph in Fig. 13.

The realisation of the proposed RaaS solution is highly dependent on the on-going software development efforts in

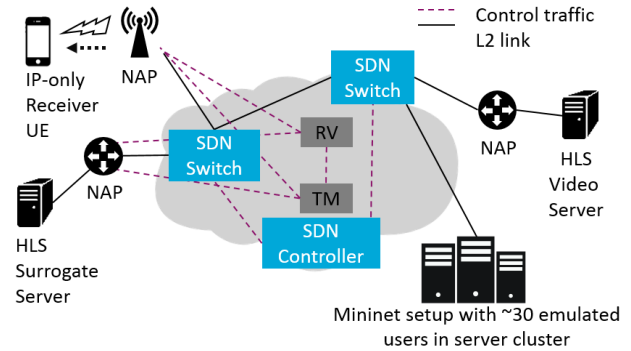


Fig. 16 Schematic set-up for MWC Shanghai 2016.

POINT, the publicly funded EU project behind Blackadder [14]. As performance benchmarks are crucial for a potential future market penetration, POINT also evaluates the designed and implemented solutions. The evaluation plan and initial validation report can be found in [4] and concludes that “both the NAP design and the SDN forwarding solution meet their key functional requirements”. A more detailed evaluation is due to be published in a later state of POINT.

6.5 Showcase at MWC Shanghai 2016

The showcase deployment described in Sect. 6 was demonstrated in a similar fashion at MWC Shanghai in 2016. A schematic set-up is provided in Fig. 16 which shows a NAP connecting an HLS video server to the ICN network (grey cloud) as well as a Mininet garden which hosts 30 clients and NAPs. One NAP was deployed at the booth in Shanghai, illustrated on the top left, and one surrogate server and its NAP at InterDigital’s office in Seoul, Korea.

The original HLS video server as well as the Mininet garden were hosted in Europe. Furthermore, several hardware SDN switches were used to interconnect ICN nodes demonstrating the successful integration of ICN-over-SDN. At the booth, a video playout including the multicast gain was demonstrated using the local UP endpoint as the trigger to request the MPD file to which the Mininet garden reacts.

The showcase demonstration also allowed turning on a surrogate server which—once being activated—provided the video playout to the Shanghai client based on a shortest path metric (number of hops in the overlay network from China to Korea will be lower than from China to Europe). The video served by the surrogate was encoded with a different banner in order to demonstrate the different playout point being chosen for it.

7. Conclusions

IP-based services are the foundation for the Internet experience that all of us enjoy nowadays and—inevitably—this will be the case for the foreseen future. Hence, improving on this experience by lowering the content delivery cost does directly impact network and content providers and indirectly also the end users by the likelihood of an improved quality of

experience. The RaaS solution presented in this paper provides such capabilities, leveraging the benefits information-centric networking architectures provide and utilising the solutions developed. This paper described the architectural changes and mechanics required to re-introduce multicast for HTTP communications and to enable surrogacy seamlessly from endpoints running a standard IP stack. The showcase deployment of 10 emulated client in an HLS scenario successfully demonstrated the multicast gain possible with the principles described in this paper which provides a significant reduction in network utilisation. Although it remains to be seen how much synchronisation of HTTP requests exists across a large video content catalogue, the straight forward and affective content delivery over the described RaaS solution makes a strong claim towards a candidate as a next generation network architecture.

Future work will focus primarily on architectural topics around supporting HTTPS, more advanced proxy rules to incorporate the diversity of the HTTP protocol combined with the increase of the Mininet deployment in terms of number of users to demonstrate the RaaS solution at much larger scale. Furthermore, a more dominant surrogacy usage is currently under development and will enable the showcase deployment to dynamically (de)place and (de)activate surrogates which is going to enrich the capabilities of the RaaS platform.

Acknowledgments

This work has been carried out through the EU-funded project H2020-ICT POINT (grant agreement No. H2020-ICT 643990) and the EU-funded H2020 project RIFE (grant agreement No. 644663).

References

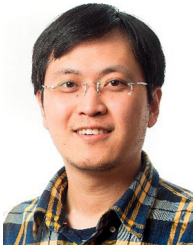
- [1] "IP Over ICN - The Better IP? An Unusual Take on Information-Centric Networking," <http://arxiv.org/pdf/1507.04221.pdf>
- [2] N. Fotiou, M. Georgiades, S. Hadjitheophanous, A. Karila, M. Al-Naday, G.C. Polyzos, S. Porter, M.J. Reed, J. Riihijärvi, S. Robitzsch, N. Thomos, D. Trossen, and G. Xylomenos, "POINT D2.1: Scenarios, Requirements, Specifications and KPIs, 1st version," <https://www.point-h2020.eu/deliverables/>, 2015.
- [3] M. Al-Naday, S. Hadjitheophanous, G. Petropoulos, M.J. Reed, J. Riihijärvi, S. Robitzsch, S. Spirou, D. Trossen, and G. Xylomenos, "POINT D3.1: First Platform Design," <https://www.point-h2020.eu/deliverables/>, 2015.
- [4] M. Al-Khalidi, M. Al-Naday, I. Doumanis, N. Fotiou, M.J. Reed, J. Riihijärvi, S. Robitzsch, and D. Trossen, "POINT D4.1: Evaluation Plan and Initial Validation Report," <https://www.point-h2020.eu/deliverables/>, 2015.
- [5] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach and T. Berners-Lee, "Hypertext transfer protocol—HTTP/1.1," IETF RFC 2616, 1999.
- [6] D. Trossen and G. Parisi, "Designing and realizing an information-centric internet," *IEEE Commun. Mag.*, vol.50, no.7, pp.60–67, July 2012.
- [7] P. Jokela, A. Zahemszky, C.E. Rothenberg, S. Arianfar, and P. Nikander, "LIPSIN: Line speed publish/subscribe inter-networking," *SIGCOMM Comput. Commun. Rev.*, vol.39, no.4, pp.195–206, 2009.
- [8] J. Tapolcai, A. Gulyas, Z. Heszbergery, J. Biro, P. Babarcsi, and D. Trossen, "Stateless multi-stage dissemination of information: Source routing revisited," 2012 IEEE Global Communications Conference (GLOBECOM), pp.2797–2802, 2012.
- [9] C. Tsilopoulos and G. Xylomenos, "Scaling Bloom filter-based multicast via filter switching," 2013 IEEE Symposium on Computers and Communications (ISCC), pp.000548–000553, 2013.
- [10] Cisco, Visual Networking Index: Forecast and Methodology, 2014–2019, White Paper, http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf, 2013
- [11] V. Jacobson, D.K. Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs, and R.L. Braynard, "Networking named content," *Proc. 5th International Conference on Emerging Networking Experiments and Technologies, CoNEXT'09*, pp.1–12, 2009.
- [12] D. Trossen, M. Sarela, and K. Sollins, "Arguments for an information-centric internetworking architecture," *SIGCOMM Comput. Commun. Rev.*, vol.40, no.2, pp.26–33, April 2010.
- [13] N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown, "Reproducible network experiments using container-based emulation," *Proc. 8th International Conference on Emerging Networking Experiments and Technologies, CoNEXT'12*, pp.253–264, 2012.
- [14] Blackadder platform, available at <https://github.com/fp7-pursuit/blackadder>, 2016.
- [15] M.J. Reed, M. Al-Naday, N. Thomos, D. Trossen, G. Petropoulos, and S. Spirou, "Stateless multicast switching in software defined networks," *Proc. IEEE International Conference on Communications (ICC)*, pp.1–7, 2016.



Sung-Yeon Kim received his B.S. degree and Ph.D. degree in Electrical and Electronic Engineering from Yonsei University, Seoul, Korea in 2007 and 2016, respectively. Since 2015, he has joined InterDigital Asia (InterDigital Communications, Asia Lab). He is a member of 5G Forum. His research interests include resource allocation (scheduling), performance analysis, distributed algorithm, and optimization in communication networks.



Sebastian Robitzsch is with InterDigital Europe working in the area of next-generation networks for 5G. He been a postdoctoral researcher at Dublin City University, Ireland, in the area of real-time data mining OSS solutions in mobile networks. In the past he has been with T-Systems, Germany; Fraunhofer FOKUS, Germany; and Nokia Research Centre, Finland, working on research issues spanning from interference and self-configuration techniques in 802.11-based multi-antenna mesh networks, heterogeneous radio access networks to system architecture design for trace analytics and recommender systems for next-generation OSSs. He received his Ph.D. from University College Dublin, Ireland, in 2013 and an M.Sc. equivalent (Dipl.-Ing. (FH)) from the University of Applied Sciences Merseburg, Germany.



Hongfei Du is with InterDigital Europe conducting the research and development activities in H2020 RIFE project on edge caching and surrogate platform. He received the B.Eng in Electronic Engineering from Beijing University of Aeronautics & Astronautics, China, in 2003. He received the M.Sc, M.Phil, and Ph.D. in wireless communication, from University of Surrey, United Kingdom, in 2004, 2005 and 2007, respectively. From 2007 to 2008, he was with CREATE-NET Research Lab, Italy, coordinat-

ing and conducting EU IST research project on software/middleware design for mobile multimedia broadcasting. From 2008 to 2009, he was with Simon Fraser University, Canada, as a research fellow working on scalable video transmission over wireless networks. In 2009, he joined Multimedia Communication Laboratory, Bell Labs, Alcatel-Lucent as a member of research staff, working on standardization, protocol design and optimization in large-scale cellular-based M2M technique and video multicast/broadcast over 3GPP LTE-A.



Dirk Trossen is a Principal Engineer at InterDigital Europe, the European branch of InterDigital Inc. His main responsibility lies in establishing the European presence of InterDigital through engagements within the EU-funded Horizon 2020 workprogramme as well as within UK-funded efforts. Dirk has more than 15 years of experience in network architectures, services and wireless technology. His main contributions can be found in the area of inter-domain networking as well as seamless handovers and new

service concepts for operators. He is currently technical leading the European efforts POINT and RIFE, performing ICN research with the goal of evolutionary introduction of ICN concepts in upcoming 5G networks. He is also an active contributor to European efforts in the 5G(PPP) space through contributions to ETP and 5GPPP whitepapers. He holds a Ph.D. degree in Computer Science from Technical University of Aachen, Germany. He published more than 75 peer-reviewed papers in international conferences and journals and has currently 32 international patents.