Object-oriented Networking: An ICN-based Architecture for the IOE

Panos Georgatsos, Paris Flegkas[1], Vasilis Sourlas[2], Leandros Tassiulas[3]
[1]CERTH-ITI
Greece
[2]University College of London
United Kingdom
[3]YALE University
United States
pgeorgat@iti.gr, pflegkas@iti.gr, v.sourlas@ucl.ac.uk, leandros.tassiulas@yale.edu

ABSTRACT: *We propose the object-oriented networking (OON) framework, for meeting the generalized interconnection, mobility and technology integration requirements underlining the Internet. In OON, the various objects that need to be accessed through the Internet (content, smart things, services, people, etc.) are viewed as network layer resources, rather than as application layer resources as in the IP communications model. By abstracting them as computing objects -with attributes and methods- they are identified by expressive, discoverable names, while data are exchanged between them in the context of their methods, based on suitably defined system-specific names. An OON-enabled Internet is not only a global data delivery medium but also a universal object discovery and service development platform; service-level interactions can be realized through native network means, without requiring standardized protocols. OON can be realized through existing software-defined networking or network functions virtualization technologies and it can be deployed in an incremental fashion.*

1. Introduction

The Internet is challenged not only by the multitude and diversity of interconnected objects (content items, sensors, controllers, services, people etc.) but also by the mobile and virtualized nature of their hosting environment. Services and applications are becoming increasingly demanding and dynamic in nature, requiring access to various sets of data, other services, controlled devices or mobile users, at Internet scale. Requirements for instant object publishing and discovery, ondemand, open and secure access are immense [1]. The advent of hardware virtualization and fast computing technologies boosts scalability and cost-effectiveness of Internet operations and services however it amplifies mobility requirements. Data, services and applications are highly virtualized, migrating within and across data centers of the same or different providers. The integration of virtual computing, storage and networking technologies at global scale through open, lightweight means has become a crucial issue.

The IP communications model is not readily fit for meeting the above challenges. Communicating objects are viewed as

application layer resources, which are accessed through specialized protocols, and networking is between network endpoints.

This design makes it hard to follow object mobility and migration across the Internet, while it can lead to confined object availability within large-scale service providers, as we witness today; which, in our view, inhibits the provisioning of Internet-scale applications.

This is a submission for the First Workshop on "Internet of Everything (IoE)".

The evolution towards an all-connected world, the Internet of Everything (IoE) [2], which is underlined by strong mobility and technology integration requirements, requires a new set of global networking abstractions and related functions, beyond those of the IP model.

Network programmability and virtualization technologies, most notably SDN [3] [4] and NFV [5], are versatile means for building networking solutions, with openness and flexibility in incorporating network functionality being their main strengths; per se they do not constitute specific network architectures. Although already applied in specific environments, especially cloud environments, Internet architectures utilizing their benefits remain an active topic of research.

The named networking nature of the ICN paradigm [6]- [10] has the potential of gracefully meeting the instant availability and generalized mobility requirements of the IoE because it can address these requirements through native network means rather than through new/add-ons in protocols at the network and/or the application layers. The unique capability of ICN compared to host-centric networking for dealing with the location and service dynamics in an IoT environment [11] is highlighted in [12]. Being content-centric, existing ICN architectures adopt a pull (request-reply) model for transferring data over the network. Their ability to support conversational and notification services is not evident. Although a number of proposals has been made (*e.g.*, [13]), especially on CCN/NDN architectures, service support in ICN is an open research issue. We propose the *object-oriented networking (OON)* paradigm to the end of providing a flexible and sustainable networking and service enablement infrastructure for the IoE. Following ICN, OON views the various objects that need to be accessed or communicate via the Internet as network layer resources identified by location-independent names. By abstracting them as computing objects, with attributes and methods, rather than as content (static or dynamic) like in existing ICN architectures, OON provides for:

• Multiple-attribute descriptive object names,

• Object discovery based on description semantics even with partially-specified names,

• Any form of object interactions (pull, push or interactive) between the methods that objects expose, and as a result,

• Dynamic in-network service (object) instantiation and execution, including chained service execution.

By placing data exchange between objects in the context of their methods, required transport and application layer interactions may not necessarily rely on specialized protocols, while service development can be facilitated. Building on its semantic richness, the design of OON allows the flexible incorporation of security-by-design solutions.

The rest of the paper is organized as follows. Section II, III and IV describe the OON framework, while Section V elaborates on OON routing aspects. Section VI discusses realization and deployment aspects and Section VII presents the OON benefits. Section VIII positions OON in the context of ICN. Finally, Section IX concludes the paper and highlights dimensions for future work.

2. OON Notions and Principles

2.1 Physical and Informational Object Forms
OON proposes a "named-object" networking model, where the term "object" refers to anything that contains data *e.g.*, documents, books, articles, pictures, videos and movies, or is able to produce or consume data *e.g.*, smart meters, sensors, various services, users and individual persons. The various data-consuming, -producing or -holding objects are seen as instances of specific well-defined classes since they can be distinguished into (sub-)classes that can be clearly and comprehensibly described, as indicated by the previous examples. This premise is in line with various on-going modeling activities like the Dublin Core Initiative [14] and the FOAF project [15].

As a result, objects can be viewed that exist in an object class-instantiated or *informational form*, comprising a set of attribute-value pairs and a list of methods, as appropriate to the class they belong. In addition, objects are viewed in their "default" *physical form* in which they actually exist in a networked environment *e.g.*, as a file, a computer process, a physical thing or a human being.

The following point is worth-making. Existing networking technologies are solely concerned with objects in their physical form, irrespective of whether they assume network locationspecific (as in IP networks) or named addressing schemes (as in ICN). In OON, we complement object physical forms with informational forms for abstracting networking at a user comprehensible level, which is beneficial from many aspects; semantic object naming and discovery, higher-level interactions and service development.

An object may exist in multiple physical forms *e.g.*, cached content items. However, it can only be associated with a single informational form, which signifies its availability -objects cannot be accessed unless their informational forms exist.

The attributes in the object informational forms are distinguished into: description attributes, presenting the main characteristics and properties of the object class; management attributes, holding the state, status and various use and usage statistics; and, relationship attributes, holding "pointers" to (the identifiers of) the corresponding object's physical form(s), perhaps other objects according to well-defined relationships.

The methods in the object informational forms correspond to the rudimentary actions that can be performed on the data that the object may hold, produce or consume, such as to push/send, pull/get or sink/consume data. For instance, a content item presents a method for sending its data. Additional methods may be provided to expose the particular higher-level communication services that a specific object class may offer *e.g.*, to chat or to talk for persons or to pause or to jump for videos. It is worth noting that security and management functionality may be embodied in the object methods and/or in other specialized, not publicly visible though, objects. Without loss of generality (see Section VI), it is assumed that the realization of the object methods is part of their physical forms.

2.2 Data and Information Networking Layers
OON encompasses two distinct networking layers: the *information networking layer* and the *data networking layer*. The data networking layer maintains and interconnects objects in their physical form and it is responsible for transferring data between objects. The information networking layer maintains and interconnects objects in their informational form and enables the discovery of objects that can be accessed at global scale.

It is noted that object discovery is based on the semantics of the attributes included in the object informational forms (meta-data), not on the data that the objects actually contain or produce. As such, OON does not obsolete the role of search engines. On the contrary, it facilitates required crawling operations, while it enhances the ability to search for everything connected as that becomes available in the Internet, including "smart things" and their data which currently are out of searching scope; these can be achieved by crawling first the information networking layer. Furthermore, OON calls for a distributed multi-polar search paradigm to the benefit of scalable global and instant search of everything.

Logically, the information networking layer lies above the data networking layer since object informational forms hold pointers to (the identifiers of) the corresponding object physical forms. The two layers can operate completely independently with the correctness of their operations depending on the consistency between the object forms in each layer. In fact, the information networking layer can operate on top of IP and ICN-based network layers (see discussion at end of Section VIII). Obviously, appropriate mechanisms should exist in order to ensure that the objects' informational forms maintain valid and up-to-date pointers to their physical forms.

In the general case, the data and information networking layers are operated by different interconnected OON domains, which may not necessarily be in 1 : 1 correspondence. Domains in different layers need not be connected, since each layer has distinct networking goals and concerns. As in today's Internet, the global topology at each networking layer cannot be known. Domains at the data networking layer are mainly characterized by their geographical span, while domains at the information networking layer are mainly characterized by the volume and kinds of objects that they hold. In analogy to IP, domains at the data networking layer correspond to NSPs, while domains at the information networking layer correspond to those running the DNS backbone. In fact, the information networking layer can be viewed as a multiple-attribute naming resolution mechanism with inherent searching capabilities as opposed to a fixed-naming resolution service, as in IP.

2.3 Object Naming

Objects are identified in their physical and informational forms by distinct identifiers: *an informational name, i-name* and a *physical-form name, p-name*, respectively.

Informational names identify objects in a descriptive manner at a user-friendly abstraction level. It is taken that objects can be categorized into certain classes (types) and each object class can be completely characterized in terms of the socalled *class-defining attributes* -with objects of the same class differing in their attribute values. Class-defining attributes, by definition, are part of the description attributes included in the object informational forms. As such, object i-names and informational forms are not different structures; i-names are included in informational forms and conversely, informational forms extend i-names with additional information. Objects are identified by their i-name in the information networking layer for being discovered.

Physical-form names identify objects from a data networking perspective. The routing functions (route dissemination, aggregation, selection) in the data networking layer operate on these names. It is noted that for routing scalability and forwarding efficiency, objects could not be identified by their i-name, since i-names do not provide for high degrees of aggregation, while their structure (components and size) are largely variable. Object p-names should be hierarchical and of fixed structure for enabling scalable and efficient routing. In addition, they should not bear network location or service technology semantics for supporting object mobility and migration, while they should be persistent for facilitating caching and multicasting within and across domains. Note that bindings for verifying object integrity and provenance could be embedded in the object informational forms, not in the p-names.

The definition of the semantics and the structure of object p-names should be seen in conjunction to their assignment process (see Section II-D) and their ability to support scalable network operations. This is left as a design choice of different OON approaches, which of course are required to inter-operate at the data networking layer.

Since object i- and p-names serve distinct purposes -i-names are used for object discovery, while p-names are used for routing data between objects- they have different characteristics. Object i-names are expressive and comprehensible, being made up of common-sense attributes, which may even be guessed if not known, whereas p-names are system-centric and generally not user-friendly. Object i-names may be supplied in a partially or loosely defined form, whereas p-names are strict in syntax and value being of no use unless they are supplied correctly in their entirety. In analogy to IP networking, object i-names correspond to URIs and p-names to IP addresses.

2.4 Overall OON Operation

First, objects need to be *instantiated i.e.*, have their physical form appeared in data networking domain(s) and *published i.e.*, have their informational form created in an information networking domain. In the general case, object instantiation and publishing may take place at the same epoch following a bottom-up, first instantiated then published, or a top-down, first published then instantiated, procedure. However, publishing may not be required for objects whose data networking name, p-name, can be made known to other objects through other specialized means than public discovery.

Objects have their informational name, i-name, defined, during publication, as part of the process of filling out their informational forms. This process may be carried out through automated and/or manual means, depending on particular (publishing and/or published) object characteristics.

Objects may be assigned their p-name by the domain holding their physical form, during instantiation, or by the overlying domain maintaining their informational form, during publishing. Evidently, this choice is of significant importance since it affects the semantics, structure and number of distinct prefixes of object p-names, therefore data routing scalability and forwarding efficiency, as well as name persistency across data networking domains. Since it touches upon business aspects, it is left as an option. In any case, OON assumes that there is a trusted organization like IANA [16], which supplies domains (either in the information or in the data networking layer) with "top level identifiers" for *prefixing (the physical forms of) objects in a systematic manner*, that is in a manner that favors aggregation per individual domain. objects are instantiated, they can exchange data with other objects, through the data networking layer, provided that their p-names are known to each other (through discovery or other means).

Once objects are published, they can be discovered by other objects based on the semantics of their attributes, through interactions with the information networking layer. Their p-name can be retrieved and subsequently, if desired, data exchange

can take place in the data networking layer. Figure 1 presents the logical architecture of OON summarizing the section.

3. Data Networking Layer

The data networking layer provides for a single network message, the "Data" message, for carrying data between objects in their physical form.

Data exchange between objects is always done within the context of their methods. The "Data" message header includes information about the calling object and its initiating method, the target object and the method to be invoked therein. This quadruplet is analogous to the source and destination pairs of IP address and port number included in the IP header. It also includes a reply-to method specified by the calling object, to which the called object should send subsequent data. The default for the calling, called and reply-to methods are the generic data-related methods -to get, send and sink data- that every object supports. Additional information may be included such as transfer priority and cumulated transfer time, as deemed necessary by specific OON approaches.
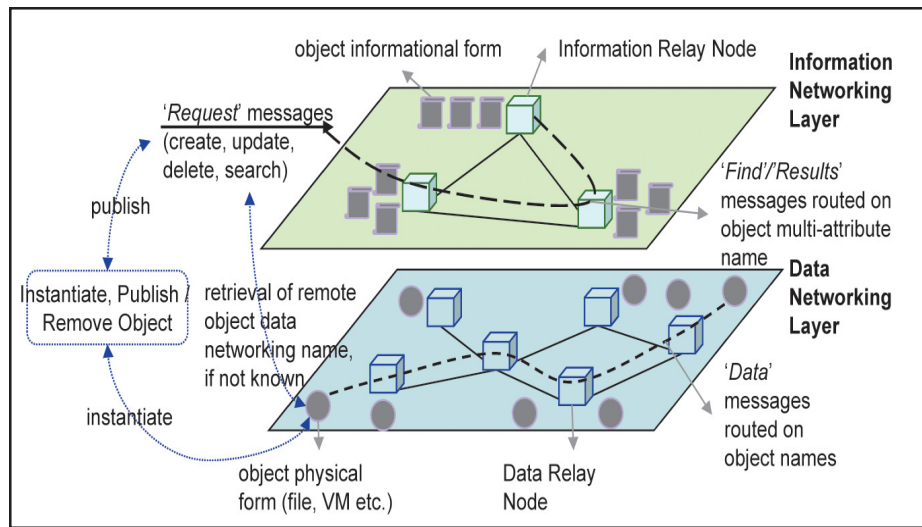

Figure 1. OON model

Generally speaking, objects can learn about the methods that other objects support by discovering the objects and "reading" their informational forms. The latter can be done manually or though automated inspection means. Note that the methods of widely used object classes may be globally known. In the following we exemplify the "Data" header information through typical data exchanges. Object methods are named as actions that can be performed to the objects than as capabilities offered to other objects. That is, a content item exposes a "send data" method than a "get data" method for enabling other objects to retrieve its data.

If a data consumer, object *A*, wishes to retrieve the data of a file, object *B*, object *A*, say its method GetDataFrom, will issue a "Data" message to object *B*, calling for its method, say SendDataTo; subsequently, for sending the data to the requester, object's *B* called method will issue "Data" messages to object *A*, calling for its method, say SinkDataFrom, or the particular method that object *A* specified in the initial "Data" message.

The voice conversation between two persons, objects *A* and *B*, involves an exchange of "Data" messages in both directions. Assuming that the two objects have agreed to converse, object *A*, say its method Talking, sends "Data" messages to object *B*, say to method Listening, specifying also that the reply data should be sent to its method, say Listening; and, similarly, in the other direction, object *B* sends its own "Data" messages to object *A*.

"Data" messages are routed on called-object information *i.e.*, on the names identifying object physical forms (p-names, see Section II-C). Calling-object information or other information could be utilized for providing differentiated routing. It is taken that object methods exist within local object scope, implying that object physical forms cannot be distributed. As such, object method information cannot be globally identifiable and therefore used for routing.

The design choice of including not only object but also method information in the data-carrying messages is made to the benefits of network performance, security and higher-level communications. From a network performance perspective, it allows for dynamic in-network service instantiation and execution, thus providing an additional mechanism for optimising network performance and QoS provision. From a security perspective, it adds extra levers. Method names, in particular the reply-to method, could be cryptographic and/or could vary during sessions for continuously asserting that data are exchanged between legitimate parties. Higher-level communications are facilitated in terms of flexibility, performance and service provisioning as there is no need to rely on specialized service protocols. Higher-level services can be considered as objects per se and therefore their interfaces -object methodscan be invoked natively by the proposed network message.

Consensus on required capabilities and interface semantics can be established significantly faster than agreements on technical specifications through explicit standardization cycles. Scalable means for inter/intra-domain routing based on physical-object names (see Section V-B), associated protocols, mechanisms and generic nodal functions are the main challenges in this layer. Caching and multicast, which are obviously facilitated by the named-object nature of OON, are not considered at a framework level; they are left as design issues to be dealt within the context of specific OON approaches.

4. Information Networking Layer

The information networking layer provides for the "xFind" and the "Results" messages, which, in the general case, are in 1-N correspondence. Additionally, it provides for interface messages corresponding to the actions and their responses that may be performed on object informational forms, namely: find the ones matching certain criteria, register a new one, modify and delete existing ones. Interface action messages translate to appropriate "xFind" messages and the produced "Results" messages to interface responses. In its find-version, the "xFind" message includes information about the object(s) of a specific class to be sought for. This information corresponds to a partially defined object informational form (and i-name, see Section II-C). Specifically, it contains pairs of the object description attributes and logical expressions for their values. The message traverses the distributed environment maintaining published objects' informational forms to the end of locating those matching the included information. As they are located, they are packaged in "Results" messages and returned to the node that issued the "xFind" message following the reverse route.

In its other versions, the "xFind" message includes the exact attribute-value pairs of the object's informational form (*i.e.*, the fully specified i-name) to be created, modified or deleted. In these cases, the "xFind" message effectively checks the absence or existence of the carried informational form, which only if so is created or modified/deleted, respectively. A single "Results" message is returned containing an affirmation of the requested action.

Generic means for scalable and efficient routing of "xFind" messages in a distributed multi-domain environment that is, for routing on multiple-attribute names with precisely or loosely specified values, is the main challenge in this layer (see Section V-A). In addition, procedures and mechanisms for aligning, merging and distributing object informational forms across domains are required. Obviously, these procedures are performed off-line at the granularity of respective agreement establishment between domains.

Mechanisms for secure object access should be incorporated. A policy-based paradigm is recommended, whereby access control policies could be defined by the objects themselves setting the viewing (of informational forms) and communication (data exchange) rights of requesting objects. The interface messages should convey an unambiguous identifier of the requesting object *e.g.*, its informational form and be time-stamped.

This layer bears its own security concerns, which are related to its operational integrity and the validity/accuracy of the maintained information. Appropriate mechanisms should be provided for ensuring that the infrastructure, interfaces and information are not compromised; such as, mechanisms for detecting false object informational forms and for avoiding malicious attacks. These security concerns are an aftermath of OON, however they should be weighed out with its benefits; after all, they are common security concerns for which solutions have already been worked out.

The above security aspects call for a logically distinct *information security layer*, where all related mechanisms could be realized. The architecture of such a security layer is orthogonal to the proposed OON framework.

5. Routing in OON

In OON, objects are identified through their i-names and pnames in two distinct naming layers (see Section II-C). Object i-names are multiple-attribute names, which are used by the forwarding functions in the information networking layer for discovering objects of interest based on description semantics (see Section IV). Object p-names are single names, which are used by the routing functions in the data network layer for exchanging data between objects (see Section III).

5.1 Routing on Object i-names

For scalable and efficient routing based on multiple-attribute names, we propose a *lexicographic partitioning-based routing* scheme, ala multiple-dimension DHT. Multiple-attribute names can be seen as points in a multi-dimensional information space, with the dimensions corresponding to the attributes. The value space in each attribute-dimension is ordered and bound (*e.g.*, A*-*Z) and as such it can be segmented. By segmenting each attribute-dimension and taking unions of the Cartesian product of the segments per dimension, lexicographic partitions can be formed. The partitions are then assigned (either individually or in groups) into suitably interconnected nodes so that lexicographic continuity between partitions in adjacent nodes is ensured. This gives rise to a network of *Information Relay Nodes (IRNs)*, each holding and managing a specific subset of the whole multiple-attribute namespace.

Depending on how partitions are allocated to IRNs and how IRNs are connected, suitable forwarding schemes on multiple attribute names can be specified, ranging from trivial "signmatching" navigation (nodes are seen as junctions and their interfaces signed exits) to more elaborate metric-oriented nexthop selection schemes. Note that in this partitioning-based routing paradigm, "topology follows routing" and not "routing follows topology" as in today's data networks. Evidently, the proposed lexicographic approach for routing in multiple-attribute namespaces scales with the number of objects since it avoids the requirement for an explicit exchange of routing information as objects are added/updated/deleted.

The multiple-attribute namespace is partitioned beforehand across IRNs and at publishing epochs, (the informational form of) an object just needs to be forwarded to the IRN assigned to maintain the particular partition to which it belongs. Similarly, at discovery or removal request epochs, that IRN needs to be reached.

The formation of lexicographic partitions and the topology of the IRN network are the free parameters for tuning to best meet the intrinsic aspects of routing in multi-dimensional namespaces. We have worked (to be published) in such schemes and the results confirm scalable and efficient operations at the expense of computing and connectivity resources.

5.2 Routing on Object p-names

OON prescribes that object p-names are hierarchical with fixed number of components. Without loss of generality it can be assumed that their structure is of the form GlobalId/LocalId, where the first part should be globally unique and is used for inter-domain routing, while the second part should be unique within the first part and is used for intradomain routing. As outlined, object p-names are assigned by providers holding the object informational or physical forms. As such, the magnitude of GlobalId is in the order of the number of such providers, not of the objects, which obviously impacts positively on inter-domain routing scalability. Routing on p-names can follow the routing schemes proposed in the literature for (single) name-based routing in ICN. In general, such schemes rely on (late) location bindings through name resolution mechanisms or on in-network information exchanges or on combinations of them [17]. Obviously, routing scalability depends on the volume and frequency of the required information exchanges, which, in addition to naming, depends on the dynamics of object introduction/ withdrawal and mobility/migration characteristics. Different routing schemes for different classes of objects, each with its own level of dynamics and metrics, may need to exist instead of one routing scheme to fit all. The object-oriented nature of OON facilitates the introduction and co-existence of per object-class routing schemes.

6. Realization and Deployment Aspects

The data and information networking layers can be realized through existing software-defined networking or network functions virtualization technologies. Each of these layers can be deployed in an incremental fashion. The information networking layer can initially be deployed on top of existing IP networks to provide for semantic object discovery and resolution to IP addresses or URIs. Subsequently, the data networking layer can be deployed, again in an incremental fashion *e.g.*, for different classes of objects. It is noted that the afore-mentioned realization technologies facilitate deployment and interoperation with IP.

As stated in Section II-A, the realization of the objects' methods is considered to be part of their physical form. Evidently, this is the case for service kind of objects. For other kinds of objects, this can be achieved through alternative means. Indicatively,

we mention: by specialized "container objects" or by transparent applications *e.g.*, proxy OON servers, in which cases they (the containers or the proxy servers) should appear in the OON object name space on behalf of the objects that they hide; or, by native means at file system or OS levels, in the end-system where the object physical forms exist. It is noted that the notion of containers is not only useful for deployment but also for scalability reasons.

Transport control logic could be provided as part of the realization of object methods or through specialized objects, which evidently should not be publicly accessible. Similarly, network and service management intelligence could be provided in the form of OON-adhering objects, without requiring standardized protocols for their interactions. Overall, OON proposes an *object-oriented communications model* advocating an open communications software market.

7. OON Benefits

By design, OON provides for seamless data transfer and semantic discovery of objects across an Internet-connected world of everything (of content, things, services and people) through common means, *avoiding interoperability problems between different naming and networking systems for different kinds of objects*. Building on the ICN paradigm (see Section VIII), its "named-object" networking nature can *inherently support mobility, migration, multicast, in-network service execution and caching*.

By distinguishing between object informational and physical forms, OON allows the *flexible accommodation of securityby-design solutions* for object verification and access control, dentity management and privacy, without burdening the naming structure used for data networking.

OON promotes a communications model, where higher level interactions may not *necessarily be based on specialized, rigid protocol stacks, but rather on a cascade of method calls*; in OON, service capabilities can be exposed in a technologyagnostic manner, may be discovered if not a-priori known and invoked by native network means. This obviously impacts positively on Internet service provisioning.

Finally, OON constitutes a *sustainable networking and service enablement infrastructure* not only for the evolving dynamics of the current Internet but also for future requirements. Required functionality, being basically software, can be introduced in the Internet as needed in the form of OONadhering objects.

8. OON and ICN

OON follows the design principle of name-based networking of ICN however it does not build around the notion of named-data (content) as existing ICN architectures (realizations of the ICN name-based networking principle) do. To us, the ICN paradigm is orthogonal to content. By bringing object-oriented principles into ICN, OON builds around the notion of named-object, generalizing the notion of nameddata.

An OON named-object comprises attributes (meta-data and content per se), a name and methods through which its content can be accessed. As such, OON differs from existing ICN architectures and brings unique benefits over them: \

• Semantically rich names; in current ICN architectures names are hierarchical or flat, not of a descriptive multiple-attribute structure as in OON that evidently bears richer semantics.

• Discovery and dynamic name construction capabilities; in current ICN architectures names should be supplied as provided by the producer/publisher, whereas in OON they may be supplied in a partially-specified form and being filled in either automatically or by the user if multiple options exist.

• Native means (no external machinery) for bridging user and network views of the world - because of the above. Note that this does not obsolete current Web search engines, which provide for deep-data-(content)- based search, whereas OON can only provide for metadata- based search.

• Native support of any service interaction model; current ICN architectures are receiver-initiated (providing for a pair of data retrieval messages, request and reply messages), whereas OON can natively support data exchange between everything connected in any possible mode -pull, push or interactive.

• Native service support; current ICN architectures may require dedicated standardized protocols at transport and application layers, whereas OON advocates a protocol stack-free communication model - required intelligence (brought as objects) and interactions (realized through method-oriented data exchange) can be developed and deployed fast.

Generalizing ICN from named-data to named-object networking, all existing ICN architectures can be mapped to OON. For example, the CCN/NDN architecture [6] maps to the data-networking layer of OON with the Interest and Data messages corresponding to the OON Data messages addressing the send-data and sink-data methods of the remote object, respectively.

Last, it is noted that the upper layer of OON, the information-network layer, that brings the three first of the above benefits, is missing from all ICN architectures. As this layer is agnostic to the underlying data networking technology, it could be integrated with existing ICN architectures, much like as with IP, for providing semantic object discovery and resolution, which is currently missing from these architectures.

9. Conclusions and Future Work

We proposed the object-oriented networking (OON) paradigm as an Internet architectural framework, for meeting current and future interconnection, mobility, migration, technology integration and service provisioning requirements. The key element of the OON design is the interconnection of the various entities that communicate or are accessed via the Internet from an informational perspective. *Everything connected is seen as an object with attributes, via which it is discovered, and methods, via which it is accessed.* This enables distributed, multi-polar search of everything based on description semantics, in-network chained service execution and the realization of higher-level services and applications by network-native means. OON unifies interconnection, discovery and service enablement across the various "Internet of" worlds of Content, Services, Things, People, which today are fragmented as they rely on diverse and often incompatible communication stacks and in some cases they are confined to specific service provider domains.

Aspects of future work relate to the development and performance evaluation of the technical challenges underlining the operation of the data and information networking layers: schema of object informational forms, naming structure(s) for data networking and associated, network topology-dependent or agnostic, information exchange procedures according to different levels of object instantiation and mobility dynamics; common machinery between multiple-attribute and single name-based routing; procedures for object instantiation and publishing; TE and self-management functions at the information and data networking layers; and, security infrastructure and mechanisms.

References

[1] Physical Web, http://google.github.io/physical-web/

[2] Evans, D. (2012). The Internet of Everything: How More Relevant and Valuable Connections Will Change the World, https://www.cisco.com/web/about/ac79/docs/ innov/IoE.pdf

[3] Software-Defined Networking: The New Norm for Networks. (2012). White paper - Open Networking Foundation.

[4] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J. (2008). OpenFlow: Enabling innovation in campus networks, *ACM SIGCOMM Computer Communication Review*, 38, April.

[5] Network Functions Virtualisation – Introductory White Paper. (2012). SDN and OpenFlow World Congress, Darmstadt-Germany. http://portal.etsi.org/nfv/nfv white paper.pdf

[6] Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N., Braynard, R. (2009). Networking named content, in ACM CoNEXT.

[7] Trossen, D., Parisis, G. (2012). Designing and Realizing an Information- Centric Internet, IEEE Communications Magazine, 50, (7), 60-67.

[8] Chai, W. K., Wang, N., Psaras, I., Pavlou, G., Wang, Ch., de Blas, G. G., Salguero, F., Liang, L., Spirou, S. (2011). An. Beben and El. Hadjioannou, CURLING: Content-ubiquitous resolution and delivery infrastructure for next-generation services, *IEEE Communications Magazine*, 49, 112–120.

[9] Ohlman, B., et al. (2009). First NetInf architecture description, April. http://www.4ward-project.eu/index.php?s=file

download&id=39.

[10] Xylomenos, G., Ververidis, C. N., Siris, V. A., Fotiou, N., Tsilopoulos, C., Vasilakos, X., Katsaros, K. V., Polyzos, G. C. (2013) *A Survey of Information-Centric Networking Research*, *In*: *IEEE Commun. Surveys*.

[11] Atzori, L., Iera, A., Morabito, G. (2010). The Internet of Things: A survey, *Computer Networks - Elsevier*, 54, 2787–2805.

[12] Zhang, Y., Raychadhuri, D., Ravindran, R., Wang, G., ICN based Architecture for IoT, IETF *Internet Draft*, ICNRG WG, December.

[13] Jacobson, V., Smetters, D. K., Briggs, N. H., Plass, M. F., Stewart, P., Thornton, J. D., Braynard, R. L. (2009). VoCCN: Voice-over Content-centric Networks, Workshop on Re-architecting the Internet (ReArch), 1–6.

[14] Dublin Core Metadata Initiative (DCMI), http://dublincore.org/

[15] The Friend of a Friend project (FOAF), http://www.foaf-project.org/

[16] Internet Assigned Numbers Authority. https://www.iana.org/

[17] Kutscher, D. (Ed.). (2013). ICN Research Challenges, Internet Research Task Force (IRTF), Work in progress, July.