Internet Engineering Task Force
Internet Draft
Intended Status: Informational
Expires: November 22, 2012

D. Papadimitriou
Alcatel-Lucent
B. Sales
Alcatel-Lucent
Th. Zahariadis
Synelixis
May 21, 2012

Internet Architecture Design Principles Evolution

draft-papadimitriou-design-principles-evolution-00

Abstract

The purpose of this draft is to extend RFC 1958 and RFC 3439 analysing the design principles that govern the Internet Architecture, evaluate how then have evolved since they were initially introduced and how we expect to evolve in the near future. We describe a number of design principle, discuss their implications on the Internet architecture, design and engineering.

The work has been based on the outcome of the ad-hoc European Commission Future Internet Architecture (FIArch) group.

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Table of Contents

1 Introduction

The Internet is the most important information exchange means nowadays. It has become the core communication environment, not only for computers but also for sensors, as well as social and human interactions. Yet, the immense success of Internet has increased the demand for both performance and functionality to fulfill the needs of, e.g. real-time applications, sensor/ad-hoc networks, and mobile networks, but without guarantees that the Internet as we know it today will be able to support them. These new demands and needs combined with the continuous expansion of the Internet (pervasive/ambient networks, vehicular networks, etc.) can up to a certain degree be addressed by means of:

i) Capacity investment: incremental infrastructure investment, e.g., more and more bandwidth in wireline, wireless and mobile networks. However, analyses have shown that increasing the bandwidth on the Internet core network will not suffice due to new qualitative requirements in, for example, highly critical services such as e-health applications, clouds of services and clouds of sensors, new social network applications like collaborative 3D immersive environments, new commercial and transactional applications, new location-based services and so on [Jacobson09] [Zahariadis11]; and

ii) Capability investment: incremental and reactive improvement of Internet protocols (and when protocol extensions are not possible complement them by means of overlays). For instance, the recent Real

Time Collaboration on the World Wide Web (RTC-Web effort) to achieve a standardized infrastructure in Web browsers on which real-time interactive communication between Web users shows that protocols hit performance walls. Hence, these limits range well beyond the classical routing and congestion control challenges of the Internet.

Hence, even if it is difficult to provide accurate timeline when applicability of classical engineering practices and associated solutions will reach their objective limits, the functional, performance as well as the structural and quality properties that the Internet architecture is expected to meet (but that cannot be resolved with current or foreseeable paradigms), lead to rethink its foundational principles.

This effort is conducted and documented in this I_D without pre-assuming a specific architectural transformation path (evolutionary or not). Design principles have played, are playing and will play a central role in the architecture of the Internet by driving most engineering decisions at conception time but also by operational decisions of ICT systems at running time. With the term design principles, we refer to:

i) a set of commonly accepted rules delineating how a designer/an architect can best structure and organize the various architectural components at conception time, and rules guiding, controlling and/or regulating a proper and acceptable behaviour at running time, and

ii) a set of fundamental and time invariant laws describing the underlying the working of an engineered artefact. Often cited as the corner stone of the Internet design compared to architectures that rely exclusively on modelling, design principles are not formally defined using a closed mathematical formulation. Classical telecommunication systems (i.e., legacy voice communication) do not consider design principles and derive their model directly from requirements.

When it comes to the design of the Internet, the formulation of its principles is a fundamental characteristic that guides the specification of its model. In analyzing the Internet design principles and their evolution, we must remember that technical change is permanent (not necessarily continuous) in the information and communication domain: "in this environment, some architectural

principles inevitably change. Principles that seemed inviolable a few years ago are deprecated today. Principles that seem sacred today will be deprecated tomorrow. The principle of constant change is perhaps the only principle of the Internet that should survive indefinitely [RFC1958].

In this context, this I_D aims to review and analyse the application of known design principles in today's and tomorrow's Internet Architecture and evaluate their potential evolution. The proposed analysis has been performed to minimize as much as possible the subjective component that arises when dealing with architectural evolution not derivable from closed mathematical formulation or proof. Analogously to [RFC3439] the ultimate goal of this I_D is not to lay down dogma about how Internet architecture and its underlying protocols should be designed. Rather, it is to convey various guidelines that have been found useful in conducting Internet-related research, and that may be useful to those designing new protocols or evaluating such designs. Finally, inline with the various architectural efforts conducted in the last two decades, it also invites the Internet community at large to initiate investigation/analysis on new design principles that will potentially drive the evolution of the Internet architecture.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [RFC2119].

1.2 Abbreviations/Definitions

It is important to define the terms we have used in our work:

o "Architecture" is a set of functions, states, and objects/ information together with their behavior, structure, composition, relationships and spatio-temporal distribution. The specification of the associated functional, object/informational and state models leads to an architectural model comprising a set of components (i.e. procedures, data structures, state machines) and the characterization of their interactions (i.e., messages, calls, events, etc.). Please note that the canonical definition of architecture includes the principles and guidelines governing their design and evolution over

time.

o "Data" to refer to any organized group of bits a.k.a. data packets, data traffic, information, content (audio, video, multimedia), etc.

o "Service" to refer to any action or set of actions performed by a provider (person or system) in fulfillment of a request, which occurs through the Internet (i.e., by exploiting data communication, as defined below) with the ultimate aim of creating and/or providing added value or benefits to the requester(s). Services are the means for users (organizations, public bodies, companies and people) to get controlled access to the available data through the Internet.

o "Resource" is any logical or physical component that may be uniquely identified. It may be service resource (e.g. a service component, a service API), or infrastructure resources (e.g. CPU, memory, network interface etc.)

o "Design principles" refer to agreed structural and behavioural rules on how a designer/an architect can best structure the various architectural components and describe the fundamental and time invariant laws underlying the working of an engineered artefact.

o "Structural and behavioural rules" is a set of commonly accepted and agreed rules serving to guide, control, or regulate a proper and acceptable structure of a system at design time and a proper and acceptable behaviour of a system at running time.

o "Time invariance" refers to a system whose output does not depend explicitly on time (this time invariance is to be seen as within a given set of initial conditions due to the technological change and paradigms shifts, the economical constraints, etc.). Robustness and longevity over time is a consequence of this time invariance.

o "Engineered artefact" is an object formed/produced by engineering.

One of the critical points that we have faced many times during our analysis has been the term "complexity". There are multiple definitions of the complexity. Within our analysis we have mainly focus on the architectural and communication complexity. Moreover, we define the following terms being used throughout this document:

o "Communication" the exchange of "data" (including both control messages and "data") between a source and a sink.

o "Communication end-point" the physical or logical source and sink of information. The communication end-point can be considered to be an application, a service, a process, a protocol, a node, a network.

o "End-to-end communication" a communication that takes place between communication end-points of the same physical or logical functional level.

o "Module" is a unity that represents functions. It could be considered as a physical or logical unity.

o "Security" is a process of taking into account all major constraints. Security includes: Robustness, Confidentiality and Integrity.

o "Robustness" is the degree to which a system operates correctly in the presence of exceptional inputs or stressful environmental conditions [IEEE-610].

o "Confidentiality" is the property of "ensuring that information is accessible only to those authorized to have access" and is one of the cornerstones of information security [ISO-27002].

o "Integrity" In literature there are multiple definitions of the integrity. Here we consider mainly the "data integrity" and "system integrity".

2 Preserving certain design principles

We start our analysis by highlighting design principles that apply to current Internet and provide arguments why they should be preserved also in the future.

2.1 Heterogeneity support principle

Since the early days of the Internet, heterogeneity is one of its major characteristics. The Internet is characterized by heterogeneity at many levels including: terminal/devices running TCP/IP stack and

intermediate nodes running routing function, scheduling algorithms and queuing disciplines applied at intermediate nodes, multiplexing of traffic (bufferless vs. elastic), traffic mix generated by a wide-variety of applications, congestion control operated at multiple spatial and temporal scales, and protocol versions and implementations.

It is important to remember that IP (as network function) has been designed as the common denominator among all data link layers themselves relying on various physical medium enabling, e.g., point-to-point (optical), multi-access (Ethernet) links. Heterogeneity is also characteristic of the organic operation of the Internet that comprises many autonomous organizations and service providers reflected by a large heterogeneity of administrative domains, each with their own separate policy (routing, traffic control, charging, etc.). As a result, the heterogeneity principle is proposed to be supported by design [RFC1958].

In the future, the heterogeneity is expected to be much higher than today. Multiple types of terminals/hosts, multiple network nodes, multiple protocols, and multiple applications will exist. Hence, the capability to support heterogeneity should remain as one of the main design principles.

2.2 Scalability and the Amplification Principle

The need to ensure scalability is of increasing importance. Scalability refers to the ability of a computational system (hardware or software) to continue to function (without making changes to the system) under satisfactory and well specified bounds, i.e., without affecting its performance, when its input is changed in size or volume or in their respective rate of variation. Accounting for the continuous expansion of the Internet (e.g., 10% annual growth rate of the number of AS, 20-25% annual increase of the number of routes), scalability is considered among the major general principles of the Internet: "All designs must scale readily to very many nodes per site and to many millions of sites" [RFC1958]. This principle refers thus to the scale invariant that the global Internet design should meet.

Scalability is also closely related with the amplification principle, which states that "there do exist non-linearities which do not occur at small to medium scale, but occur at large scale" [RFC3439]. In other words, "in many large interconnected networks, even small things can and do cause huge events; even small perturbations on the input to a process can destabilize the system's output", and "the design engineer must ensure such perturbations are extremely rare" [RFC3439].

The number of devices with Internet access (e.g., PCs, laptops, smart phones), communication nodes (e.g., home, access, edge and core routers), autonomous systems, applications in the Internet is expected to significantly increase. Moreover, the direct interconnection of sensor networks with the legacy Internet will exponentially increase the number of Internet nodes. If one sees the Internet currently comprising three level of tiers, extension of the Internet at its periphery could expectedly lead to a fourth tier which would have a fundamental impact on the properties of the routing system. As a result, we believe that scalability is among the major design principles that should govern the architecture of the Internet, while the amplification principle would become more prominent as the scale of the Internet increases.

2.3 Robustness principle

The robustness principle was based on the condition that each protocol implementation must interoperate with other created by different individuals. As there may be different interpretations of the same protocol, each one should "be liberal in what you accept, and conservative in what you send." The fundamental objective of this principle was to maximize interoperability between network protocol implementations, particularly in the face of ambiguous or incomplete specifications. Focusing on the programmable part of the Internet (being software or some programmable components), "Software should be written to deal with every conceivable error, no matter how unlikely. In general, it is best to assume that the network is filled with malevolent entities that will send in packets designed to have the worst possible effect" [RFC1122]. This assumption leads to suitable protective design, although the most serious problems in the Internet have been caused by un-envisaged mechanisms triggered by low-probability events! In the future, the Internet is expected to handle broader spectrum of applications including mission and time critical applications, related with, e.g., health, energy, robotic, transport/logistic (thus well beyond multimedia for entertainment).

As a result, part of the robustness principle that covers issues related to minimizing the malfunction, uninterrupted operation and interoperability, remains unchanged. As it is later analysed and argued, the robustness principle should be extended/ adapted to additional cover security issues.

2.4 Loose Coupling principle

Loose coupling appears to be a necessary condition for a well-structured system and a good design as i) it simplifies testing and troubleshooting procedures because problems are easy to isolate and unlikely to spread or propagate, ii) combined with high cohesion, it

supports the general goals of high readability and maintainability, and iii) it minimizes unwanted interaction among system elements. In addition, tightly coupled systems are likely to have unforeseen failure states and implies that the system has less flexibility in recovering from failure states. For these reasons, this design principle shall be preserved and even reinforced as a result of the increasing importance of the availability objective [FIArch11].

Nevertheless, recent evolution shows that loose coupling can also increase difficulty in maintaining synchronization among diverse components within a system when a higher degree of element interdependence is necessary. Hence, loose coupling is important but it would be appropriate to consider that under stress conditions, higher cohesion should be possible for proper functionality.

2.5 Locality Principle

The locality principle has played a very important role in computer design, programming and the Internet the last decades. Following the principles of spatial and temporal locality, recent computer systems have pushed cache memory to higher levels in the computer systems but the essence remains the same: reflect the chosen methods for using the principles of spatial and temporal locality. In this context, the principles of spatial and temporal locality will have to be extended to distributed computing systems and to the higher layers space of distributed application architectures. On the other hand, locality will play a fundamental role in self- stabilizing distributed systems by ensure sub-linear stabilization with respect to the number of local system components and interactions among components.

Based on the above considerations, the locality principle plays an important role and should be preserved, while its scope should be extended to cover additional roles in distributed systems and distributed application architectures.

3. Evidences for augmenting/adapting certain design principles

In this section we highlight design principles that have been described and still apply to the Internet architecture. Yet, we challenge that they should be adapted in order to address the evolving design objectives of the Internet.

3.1 Keep it simple, but not "stupid" principle

As already explained, one of the main design principles of the Internet was the approach to keep things simple (the term things refers here in particular to protocols and intermediate systems). If there were many ways to do the same thing, one should choose the

simplest one [KISS]. This common sense engineering principle continued requirement to make the usage of network functionality simple and robust, but more processing logic is needed in order to achieve the expected functionality.

In Internet design, the complexity belongs at the edges, and the IP layer of the Internet remains as simple as possible. Complex systems are generally less scalable, reliable and flexible. Architectural complexity implies that in order to increase the reliability it is mandatory to minimize the number of components and their interactions in a service delivery path, where the service delivery path can be a protocol path, a software path, or a physical path. However, this principle has already been challenged. Complex problems sometimes require more elaborated solutions and multidimensional problems such as the Internet architecture will be providing non-trivial functionality in many respects. The general problem can be seen as follows: complexity is a global measure of the architecture structure and behaviour. In that respect, arbitrary lowering complexity (over space) might result in local minimum that may be globally detrimental. Hence, the challenge becomes to determine the best placement and distribution of functionality that would globally minimize the architectural complexity. In turn, scalability and simplicity should be handled as strongly interconnected first priority principles of the Internet architecture.

3.2 "Minimum Intervention" Principle

The principle of minimum intervention states that: "To minimize the scope of information, and to improve the efficiency of data flow

through the Encapsulation Layer, the payload should, where possible, be transported as received without modification" [RFC3439]. The minimum intervention principle is critical to maintain and preserve data integrity and to avoid useless intermediate information message or packet processing.

Deep Packet Inspection (DPI), Network Address Translation (NAT) and network coding provide three good examples of detrimental intermediate in-band processing of packet flows (note: the two first are also good examples of locally deployed artefact to selfishly compensate delays in adoption of global solutions, e.g., IPv6). Moreover, in some cases, it directly conflicts with the simplicity principle and the complexity minimization principle; e.g., in sensor/machine-to-machine and ambient/pervasive networks where communication gateways (store-and-forward) and actuators meant to enable communication between networks by offloading capabilities that would be costly -or sometimes even impossible- to support on sensors.

As a result, the minimum intervention principle should be relaxed to enable wider-variety of intermediate processing at the necessary condition it decreases global complexity.

3.3 Robustness principle

Over recent years, in order to increase robustness and system reliability, some have advocated to transform this fundamental principle from "be liberal in what you accept, and conservative in what you send" into "be conservative in what you send and be even more conservative in what you accept from others". However, adoption of this approach would result into dropping a significant level of interoperability between protocol implementation.

Nevertheless with respect to security, this design principle leads to weak security architecture thus requiring adaptation. Indeed, "it is best to assume that the network is filled with malevolent entities that will send in packets designed to have the worst possible effect. This assumption will lead to suitable protective design" [RFC1122].

Henceforth, we argue that the robustness principle should be adapted to incorporate a self-protection structural and behavioural principle

(coordination of the local responses to external intrusions and attacks including traffic, data and services trace back that would enforce in turn accountability) as well as confidentiality, integrity and authentication should be inherently offered to information applications and service.

3.4 Modularity & Adaptability Principle

Current communication systems are designed as a stack of modules structured by static and invariant binding between layers (modules) that are specified at design time. Indeed, when they were developed CPU and memory were scarce resources and the expected uniformity of their utilisation (computing machine interconnection) lead to a design optimizing the cost/performance ratio at design time.

Moreover, [RFC1122] also defines adaptability as a major design principle: "Adaptability to change must be designed into all levels of Internet host software. As a simple example, consider a protocol specification that contains an enumeration of values for a particular header field -- e.g., a type field, a port number, or an error code; this enumeration must be assumed to be incomplete. Thus, if a protocol specification defines four possible error codes, the software must not break when a fifth code shows up." The second part of the principle is almost as important: software on other hosts may contain deficiencies that make it unwise to exploit legal but obscure protocol features. A corollary of this is "watch out for misbehaving

hosts"; host software should be prepared, not just to survive other misbehaving hosts, but also to cooperate to limit the amount of disruption such hosts can cause to the shared communication facility.

Nowadays, looking at current evolution with i) repetition of functionality across multiple layers, e.g., overlays that allow carrying TDM over IP over Ethernet over IP/MPLS, emphasize the need to define common patterns, monitoring modules repeated over multiple layers (which then requires to recombine information in order to be semantically interpretable) as well as security components each associated to a specific protocol sitting at a given layer (which result into inconsistent response to attacks), ii) as part of the same layer, the proliferation of protocol variants all derived from a kernel of common functions/ primitives, iii) the variability of external and internal events that communication systems have to cope

with emphasize that the cost/performance objective to be met by communication systems can vary over time (thus messages would be processed by variable sequence of functions determined at running time), iv) the increasing heterogeneity of environments where communication systems are involved emphasize that some of these functions may be more or less elaborated, and v) Increasing heterogeneity of running conditions as well as increasing occurrence of unexpected events leads to i) consider modules connected by means of realization relationships that supply their behavioural specification, ii) distinguish between general and specialized modules (inheritance), and iii) enable dynamic and variable binding between the different modules such that the sequence of functions performed is specified at running time.

This being said, in the current architecture, the transport module is not complete and thus not modular. Indeed the transport address depends on the IP address and more generally its usage relationship does exclusively depend on the existence of other modules in the stack: one can't replace or use a TCP stack without knowledge of how it will impact operations of other modules.

3.5 Polymorphism principle (as extension to the modularity principle)

Polymorphism (ability to take on different forms) in computer science/programming space applies to data (generalized data type from which a specialization is made) or functions (function that can evaluate to or be applied to values of different types). It enables to manipulate objects of various classes, and invoke methods on an object without knowing that object's type.

The introduction of polymorphism principle is driven by the motivation to make use of this fact to make our architecture simpler. In many cases, the modularity and layering principles have been the

driving principles for both communication protocols and software implementations. This principle has led to faster deployments, but suboptimal solutions; as such these principles have been challenged in many cases, especially in environments where functions of each layer needs to be carried out completely before the protocol data unit is passed to the next layer.

In this context, polymorphism enables to manage and operate first

class objects belonging to different kinds of classes (which within a hierarchy often share common methods and attributes) while providing the ability for a super-class to contain different objects of a subclass type at different points in time. In turn, this allows i) for objects of different classes to respond differently to the same function call thus results in different functionality being executed for the same method call, and ii) for run-time (dynamic) instead of compile-time (static) binding.

Henceforth, the introduction of polymorphism would enable the same abstract and autonomous loosely coupled components/objects to have different functional and non-functional behaviour under different environments or circumstances. The question remains open though as how to parameterize these environmental variables and whether this parametrization could be performed through distant exchanges (remotely) which would turn this principle close to the concept envisaged by active networks in the late 90's.

3.6 Unambiguous naming and addressing principle

As stated in [RFC1958], the Internet level protocol are and must independent of the hardware medium and hardware addressing. This approach allows the Internet to exploit any new digital transmission technology of any kind, and to decouple its addressing mechanisms from the hardware. It allows the Internet to be the easy way to interconnect fundamentally different transmission media, and to offer a single platform for a wide variety of Information Infrastructure applications and services.

Concerning name and addressing, the following augmentations are considered using [RFC1958] as starting point:

o Avoid any design that requires addresses to be hard coded or stored on non-volatile storage. When this address this is an essential requirement as in a name server or configuration server a discovery process is recommended. In general, user applications should use names rather than addresses. In that respect, the transport layer address should be decoupled from any locator and use space invariant identifiers associated to the communication end-point. In turn, this would facilitate dynamic multi-homing, TCP connection continuity

(required for mobility) and more generally misuse of IP addresses.

o A single and common naming structure should be used.

o LOC/ID separation (initially proposed by the NIMROD effort): resulting from the overload of IP address usage, upper layer protocols must be able to determine end-point identifiers (ID) unambiguously, and make use of locators (IP addresses) strictly for end-to-end routing (processing at intermediate nodes) must be the same at start and finish of transmission. This separation involves the need to provide the capability for mapping (or resolving) identifiers to locators at the end-points. Both Identifiers and Locators must be unambiguous and be unique within any scope where they may appear.

In the future, it is foreseen that not only the end-points (ID) and their attachment points (LOC) need to be unambiguous and unique within the scope in which they appear and are used, but also the objects, e.g., data files/streams, software components, etc. At the end, in most cases, the user is not willing to access a specific server, but the objects that this server hosts or offers. If exactly the same data (e.g. content, type, quality, security, etc.) and/or associated service (i.e., functional and not functional matching) can be provided in another way (e.g. from another server or method), it is also acceptable and in many cases even preferable if the actual quality is better (or cost lower).

Moreover, the current ID/LOC approach only deals with hosts and can not provide a method to ensure that an entity is the one claiming to be or, even worse, they disclose a fixed identifier that can be easily traced by any other network element to know the operations that an entity performs, thus violating its privacy.

In near future, naming and addressing as a design principle should be extended to unambiguous identify hosts, resources, data, services.

3.7 Extending the end-to-end principle

Historically, the "end-to-end principle" has been one of the most controversial issues in the Internet innovation. Many experts in the area insist that the "end-to-end" principle is still valid as it applies as the communication is divided at autonomous legs. However, the clear definition of communication end-points becomes more and more complex to delimit, as middle boxes and application layer gateways are deployed at the edges of networks.

AAnother challenge concerning this principle is that IP overlay applications such as IP multicast and mobile IP (MIP), require that

specific functionality (RP in ASM, and Home Agent in MIP) is supported and provided by (at least some)intermediate nodes. It is important to notice though that some of these functions and their spatial location (in the end-to-end communication chain) are purely driven by arbitrary choices, e.g., Proxy MIP for mobility management or delayed migrations, e.g., NAT instead of rolling out IPv6. Another challenge comes from the Internet of Things/Smart objects communication, where the end-to-end communication may be significantly modified by intermediate gateways and sensor networks sink nodes.

It is also well perceived that for many modern applications (e.g. mobile applications, distributed searching, certain aspects of collaborative computing) maintaining state information within the network may now be desirable for efficiency if not overall performance effectiveness. To argue today that the only stateful elements that may be active in the Internet environment should be located at the edges of the Internet is to ignore the evolution of software and other technologies to provide a host of services throughout the Internet [WGIG04].

Finally, as stated in the [FIArch] and further analyzed in [RFC6077], support of congestion control cannot be realized as a pure end-to-end function: congestion is an inherent network phenomenon that in order to be resolved efficiently require some level of cooperation between end-systems and the shared communication infrastructure. Instead of placing specific functions in specific positions either at end systems or on routers in the network core, these functions must be allowed to be deployed anywhere they are needed [RFC3234].

As a result, we believe that motivations to "update" or augment this principle increase; however even if this principle is challenged, due to the heavy consequence in terms of scalability, survivability and robustness on the Internet at large departing from this principle remains open.

4. Conclusions - Evidences of emergence of new seeds

In this draft, we have analyzed the evolution of existing design principles and evaluate their potential evolution. From this analysis, we have determined through qualitative but also (and when available quantitative arguments) the design principles that should be preserved, adapted or even augmented.

Yet, evidences show that the Internet (and its architecture) progressively evolves from a pure network architecture to an

ecosystem interconnecting resources of any type(forwarding, computing and storage and/or their combination) and any type of content or

information and be extended to even socio-economic dimensions. Realising such Internet Architecture ecosystem involves new design principles that go well beyond the networking primitives. In this context, new design principles will progressively appear and shape the evolution of the Internet Architecture. However, before being in a position to formulate such principles, one has to explore a multitude of seeds proposed by various architectural work and efforts conducted during the last two decades (some of which are documented in [Papadimitriou12]). A seed for a new design principle refers to a concept or a notion at the inception of a well formulated design principle. The term seed acknowledges that i) formulating principles is a complex exercise, ii) research is still ongoing in proving their value and utility (some of our analysis and exploitation of research results may not be mature enough) but also their impact, and iii) the proposed seeds may not be flourishing (a lot of proposal came in and very few will materialize).

The risk is not negligible though that without well formulated and commonly accepted design principles, evolution might lead to hamper the commonality and the genericity of a system that is inherently de-centralized but also organically participative. We consider thus that initiating systematic investigation/analysis on new (seeds of) becomes crucial as the diversity of protocol design choices and decisions in such environment (that would combine more dimensions than network connectivity functionality) becomes such that it may even non-voluntarily harm interoperability

5. Acknowledgements

This draft has been based on the work from the EC Future Internet Architecture (FIArch) group. The work leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) by a number of projects, including FP7-ICT-COAST, FP7-ICT-REVERIE, FP7-ICT-EINS, and FP7-ICT-EULER.

6. References

[Clark05] D.D. Clark, J. Wroclawski, K.R. Sollins, R. Braden, Tussle

in Cyberspace: Defining Tomorrow's Internet. IEEE/ACM Trans. Networking, Vol.13, Num.3, pp.462-475, June 2005.

[Clark88] D.D. Clark, The design philosophy of the DARPA internet protocols, ACM SIGCOMM Computer Communication Review, Vol.18, No.4, August 1988, pp.106-114.

[Conti04] M. Conti, G. Maselli, G. Turi, and S. Giordano, "Cross-Layering in Mobile Ad Hoc Network Design," IEEE Computer, Special issue on Ad Hoc Networks, February 2004.

[Dening05] Peter J. Denning, The locality principle, Communication of the ACM, Vol.48, No.7, July 2005, pp.19-24.

[Feldman07] A. Feldman "Internet Clean-Slate Design: What and Why?" ACM SIGCOMM Computer Communications Review, Vol.37, No.3, July 2007, pp.59-64.

[FIArch11] EC FIArch Group, "Fundamental Limitations of current Internet and the path to Future Internet," March 2011.

[Fogel66] L.J. Fogel, A.J. Owens, M.J. Walsh, Artificial Intelligence through Simulated Evolution, John Wiley, 1966.

[Goldsmith02] A.J. Goldsmith, S.B. Wicker, "Design challenges for energy-constrained ad hoc wireless networks," IEEE Wireless Communications Magazine, Vol.9, No.4, 2002, pp.8-27.

[Haapola05] J. Haapola, Z. Shelby, C. Pomalaza-Raez, P. Mahonen, "Cross-layer energy analysis of multi-hop wireless sensor networks," 2nd European Workshop on Wireless Sensor Networks (EWSN), Istanbul, Turkey, 2005.

[Hakala06] I. Hakala, M. Tikkakoski, "From vertical to horizontal architecture - a cross-layer implementation in a sensor network node," InterSense'06, Proc. of 1st International Conference on Integrated Internet Adhoc and Sensor Networks, Nice, France, May 2006.

[Hilborn04] R. Hilborn, "Sea gulls, butterflies, and grasshoppers: A brief history of the butterfly effect in nonlinear

dynamics", American Journal of Physics, Vol.72, No.4, pp.425-427. Bibcode 2004, doi:10.1119/1.1636492.

[IEEE-610] IEEE Std 610.12.1990 - IEEE Standard Glossary of Software Engineering Terminology.

[ISO-27002] International Organization for Standardization (ISO), "Information technology. Code of practice for information security management," ISO 27002, 2005 (replaced ISO-17799).

[Jacobson09] V.Jacobson, D.Smetters, J.Thornton, M.Plass, N.Briggs, R. Braynard, "Networking Named Content," Proceeding of ACM CoNEXT 2009. Rome, Italy, December 2009.

[Johnsson03] K.B. Johnsson, D.C. Cox, "An adaptive cross-layer scheduler for improved QoS support of mixed data traffic on wireless data systems," in: Vehicular Technology Conference, 6-9 October 2003, pp.1618-1622.

[Kempf04] J.Kempf, Ed., R.Austein, Ed., "The Rise of the Middle and the Future of End-to-End: Reflections on the Evolution of the Internet Architecture", IETF, RFC 3724, March 2004.

[KISS] "Keep it Simple Stupid". The Jargon File, version 4.4.7.

[Liu05] Y. Liu, H. Zhang, W. Gong, D. Towsley "On the Interaction Between Overlay and Underlay Routing," Proc. IEEE INFOCOM 2005.

[Papadimitriou12] D. Papadimitriou, Th. Zahariadis, P. Martinez-Julia, I. Papafili, V. Morreale, F. Torelli, B. Sales, P. Demeester, "Design Principles for the Future Internet Architecture", Lecture Notes in Computer Science, Vol.7281, May 2012, pp.55-67.

[Prasad08] R. Prasad "A Perspective of Layerless Communications," Wireless Personal Communications, 2008, pp.95-100.

[RFC793] J. Postel, "Transmission Control Protocol", IETF, RFC 793, September 1981.

[RFC1122] R. Braden, Ed., "Requirements for Internet Hosts --

Communication Layers", IETF, RFC 1122, October 1989.

[RFC1631] K. Egevang, P. Francis, "The IP Network Address Translator (NAT)," IETF, RFC 1631, May 1994.

[RFC1925] R. Callon, "The Twelve Networking Truths," IETF, RFC 1925, April 1996.

[RFC1958] B. Carpenter, "Architectural Principles of the Internet," IETF, RFC 1958, June 1996.

[RFC2775] B. Carpenter, "Internet Transparency", IETF, RFC 2775, February 2000.

[RFC3234] B. Carpenter, "Middleboxes: Taxonomy and Issues," IETF, RFC 3234, February 2002.

[RFC3439] R. Bush, D. Meyer, "Internet Architectural Guidelines," IETF, RFC 3439 (updates RFC 1958), December 2002.

[Saltzer84] J.H. Saltzer, D.P. Reed, and D.D. Clark, "End-To-End Arguments in System Design", ACM Transactions on Computer Systems (TOCS), Vol.2, No.4, November 1984, pp.277-288.

[Stevens74] W. Stevens, G. Myers, L. Constantine, Structured Design, IBM Systems Journal, Vol.13, No.2, 1974, pp.115-139.

[WGIG04] "The End-End Principle and the Definition of Internet", Working Group on Internet Governance (WGIG) Contribution of Corporation for National Research Initiatives. Prepared by: Patrice A. Lyons, November 10, 2004.

[Willinger02] Walter Willinger and John Doyle, "Robustness and the Internet: Design and evolution", 2002

[Zahariadis11] Th. Zahariadis, D. Papadimitriou, H. Tschofenig, S. Haller, P. Daras, G. Stamoulis, M. Hauswirth, "Towards a Future Internet Architecture," Book Chapter, "Future Internet Assembly," Lecture Notes in Computer Science, Vol.6656, Springer, 2011, pp.7-18.

Authors' Addresses

Dimitri Papadimitriou
Bell Labs, Alcatel-Lucent, Belgium
EMail: dimitri.papadimitriou@alcatel-lucent.com

Bernard Sales
Bell Labs, Alcatel-Lucent, Belgium
EMail: bernard.sales@alcatel-lucent.com

Theodore Zahariadis
Synelixis, Greece
EMail: zahariad@synelixis.com