Towards A Future Internet Architecture

Norbert NIEBERT[1], Henrik ABRAMOWICZ[2], Martin JOHNSSON[2], René REMBARZ[1], Holger KARL[3] ,Christopher FOLEY[4,] Pedro A. ARANDA GUTIÉRREZ[5]

[1]*Ericsson Research, Corporate Unit, Ericsson GmbH, Ericsson Allee 1, 52134 Herzogenrath, Germany, norbert.niebert@ericsson.com*

[2]*Ericsson Research, Corporate Unit, Ericsson AB, 164 80 Stockholm, Sweden,* [3]*University of Paderborn, Germany,* [4]*Telecommunications Software & Systems Group, Waterford Institute of Technology, Waterford, Ireland,* [5]*Telefónica, I+D, 28043 Madrid, Spain*

Abstract: In this paper we describe the architectural approaches taken in the 4WARD project to address the challenges of the network of the future. Our main hypothesis is that the Future Internet must allow for the rapid creation of diverse network designs and paradigms and must also support their co-existence at run-time. A novel network design process and deployment architecture is needed that enables new interoperable network architectures on top of established infrastructure. To enable this flexibility, networking resources need to be described in a uniform way. We outline the generic path architecture in this paper as our approach to this challenge. Moreover, the Internet's focus on interconnecting hosts and delivering bits has to be replaced by a more holistic vision of a network of information and content. This is a natural evolution of scope requiring a re-design of the architecture. To make this flexibility economically viable, a radically new management architecture has to be designed. We describe all these architectural elements that together form a framework for diversified but interoperable networks of the future.

Keywords: Future Internet, Network Architecture, Network Virtualisation, Self-Management, Information-centric Networking.

1 Introduction

The discussion on the "Network of the Future" is gaining in intensity due to increasing concerns about the inability of the current Internet to address a number of important issues affecting present and future services and to the impetus provided by "clean slate design" research initiatives launched in the US (e.g. around the GENI programme), Europe (e.g. within the Future Internet Assembly), and Asia (e.g. in the AKARI project). Many problems with the current network architecture have been recognized for a long time but have not received a satisfactory solution (see e.g. [1]). The 4WARD project [2] is developing a consistent set of new technologies and architectural concepts that have the potential to become cornerstones of a Future Internet. It is addressing the deficiencies of the current Internet at its very heart, the network layer itself, and beyond.

This paper integrates the project results from a high-level architectural view, focussing on a new network design process, inherent self-management of the network, virtualisation of networks, generic connectivity, and finally the networking of information. The elements of the 4WARD architecture framework as described here are designed for the extension to a whole family of networks. These networks inherently satisfy security, mobility, and quality of service, and interoperability requirements which are necessary for a global deployment – the ultimate goal of an architecture for the network of the future.

The outline of this paper is as follows: In the next Section 2 we first give a short summary of our motivation to undertake this work. We continue with an elaborate description of the key features and aspects of the architecture that is emerging at this stage. An outlook and conclusions are given at the end of the paper.

2 Motivation for a new architectural approach

The Internet was initially developed for a limited number of trusted nodes interconnected by copper transmission for distributed applications like file transfer and message exchange.

Its initial architecture was essentially simple but open for new applications. Its evolution has led to a tremendous success – the Internet as we know it today. Its pragmatic standardisation process has led to a situation where the Internet has always coped with new challenges both by means of new application demands, new link layers, notably wireless and fibre and new core routing capabilities. The price for this flexibility is increasing cost for operating Internets and the difficulty to do complete overhauls of the architecture as can be seen by the slow IPv6 introduction. The Internet has reached a state of high complexity with regard to interoperability, configuration, management and also a high degree of vulnerability in an untrustworthy world.

Within the research community the need for change is largely acknowledged although there is not yet agreement on how this should come about. Some propose a *clean slate* approach, which aims at complete redesign of the Internet with new requirements in mind; others are advocating an evolutionary approach [8].It seems likely that both approaches will exist in parallel and coexist over a longer transition time. This paper looks especially into the architectural propositions of a modular clean-slate approach that nevertheless could lead to a directed evolution of the Internet. The analysis done by the 4WARD project manifested in a set of technical requirements [3] have fostered a new architectural proposal that is based on a more elaborate and complete network view, bringing innovation back into the network architecture itself.

To achieve this, it must be possible to design and deploy network architectures without replacing infrastructure, it must be possible to quickly respond to new application demands leading to enhanced networking functionality, but also ensuring a scalable interoperability solution. Is such an approach that clearly addresses the fundamental issues with the IP architecture feasible? In 4WARD we are creating both a framework for this agile networking view and an example for a completely new network architecture. This architecture incorporates the features and concepts of an information-centric paradigm to prove the feasibility and also to create a new and attractive new network offering. In the next section we will outline our architectural approach.

3 Key concepts, features and aspects of the 4WARD architecture

This section describes concepts, features, and different aspects that will make up the cornerstones of a 4WARD architecture. These aspects result in the 4WARD Architecture Framework. It allows designing interoperable network architectures based on a set of new architectural constructs, enabling rapid deployment of network functionality. In-Network Management self-manages functionalities even across networks. Network virtualization gives the capability to virtualize the resources of the underlying physical infrastructure (thus creating VNETs), improving resource efficiency and easing migration problems. The Generic Path concept defines new constructs for how to flexibly establish and manage connectivity, mobility, and technology dependencies, e.g., a wireless medium. And finally, the Network of Information builds a new infrastructure to manage dynamic information objects. Overall, the 4WARD architecture addresses the key challenges of dynamic and scalable internetworking.

3.1 The Architecture Framework

The Architecture Framework must provide ways to (i) guide the Network Architect to allocate the required network functionalities and (ii) assure the interoperability within families of network architectures.

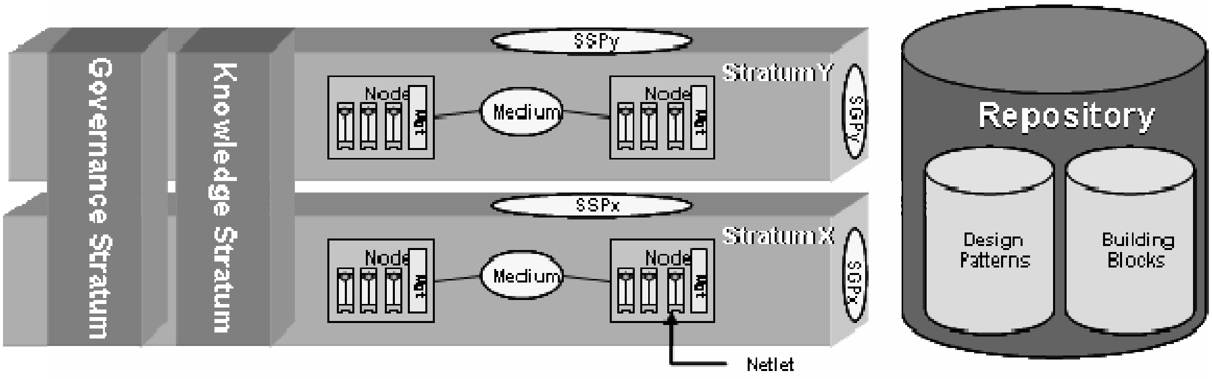

Figure 1: High-level view of 4WARD Architecture Framework

As can be seen in Figure 1, the following main components constitute this framework:

- A Stratum is modelled as a set of logical Nodes that are connected through a Medium that provides the means for communication between the Nodes inside this stratum. This stratum encapsulates functions that are distributed over the nodes. These functions are provided to other strata through two well known interfaces (that can be also distributed over the nodes): The SSP (Service Stratum Point) that provides the services to the other strata located on top of the respective Stratum and to the vertical strata. Figure 1 shows Stratum Y using the services provided by Stratum X through SSP_X. The SGP (Service Gateway Point) offers peering relations to other strata of the same type.
- Strata can manage themselves. For example, when a routing service stratum is deployed, it organizes itself onto the physical infrastructure. The deployment will be in accordance with the specification of the logical nodes and the medium of the stratum, taking then into account the topology, capabilities, and resource status of the nodes and links in the physical infrastructure.
- Horizontally stacked strata (as shown in the middle of Figure 1)figure are related to the transport and management of data across networks. Within such strata, Netlets can be considered as containers for networking services. They consist of functions/protocols inside a Node that are needed to provide the services. By virtue of containing protocols, Netlets can provide the Medium for different Strata, i.e. inside the same Netlet there could be functionalities that are related to different strata. Figure 1 shows such Netlets implementing media for different strata inside the same node.
- The two vertically oriented strata provide Governance and Knowledge for an entire network (i.e. a set of horizontal strata). The Knowledge Stratum provides and maintains a topology database as well as context and resource allocation status as reported by a horizontal stratum. The Governance Stratum uses this information, together with input provided via policies, to continuously determine an optimal configuration of horizontal strata to meet the performance criteria for a network. The Governance Stratum also establishes and maintains relations and agreements with other networks.
- The Repository contains the set of Building Blocks and Design Patterns for the composition of functionalities (i.e. to construct the strata and the netlets) for specific network architectures, including best practices and constraints to ensure interoperability between network architectures.

3.2 In-Network Management

Management of today's networks was an afterthought. It has been added on, literally on top of today's network with one centralized entity controlling everything. This has led to numerous difficulties; the ability to scale up to very large network sizes, the amount of human intervention needed and it’s associated cost, and the poor response times to constantly changing networks.

In-Network Management (INM) addresses these challenges through a clean slate approach. The INM paradigm can be interpreted as pushing management intelligence into the network, and not just into one network node but distributing the management logic across all nodes. Therefore the network becomes more intelligent. One of the major benefits of this will be the support of a management by objective approach, which will simplify the complexity of management operations for the future Internet.

Realizing the design of an INM solution has resulted in three main components; (1) an INM framework, (2) algorithms for decentralized real-time monitoring, anomaly detection and situation awareness, (3) network-wide self-adaptation schemes. The INM framework proposes a distributed architecture which brings the management functionality closer to the services. The monitoring algorithms provide the necessary input to the decision-making processes of network management and their results feed into the self adaptation schemes. The self adaptation looks at taking corrective management actions for the purpose of recovering from a fault, avoiding a predicted fault, or optimizing the network operation, by changing the network configuration, the network setup, or resource allocation.

The INM framework proposes a number of key architectural elements, two of which are most prominent: Management Capabilities (MC) and Self Managing Entities (SE). The MCs are the management logic and they reside inside the SE. Each SE is associated with a service. The SE has a number of mandatory properties: *self-knowledge, self-management*, *self–protection*, *composability and*, *auditability*. These self-* properties allow for automated processing and the reduction in the need for manual intervention. The management capabilities are the key enablers of these properties. The MCs collaborate with each other in order to achieve this autonomous behaviour.

The framework introduces the concept of a ‘level of embedding’ for management capabilities. At the lowest level, an MC can be *inherent* to the service which its SE is realizing. The other levels of embedding are: *integrated*, *separated* and *external*. The goal is to push the MCs down to their most appropriate level of embedding. In doing this, the management is as close as possible to the service it’s related to. The level of embedding also allows for an MC to be decoupled from a service if this is the optimal solution.

The elements and mechanism of the INM framework will complement the other features developed in 4WARD, namely Network Virtualization [3.3], Generic Path [3.4] and, Network of Information [3.5]. The MC construct allows the encapsulation of a piece of management logic which can be tightly or loosely coupled with the service, e.g. management of a data flow inside a Generic Path or the specific management requirements of a Network of Information object.

The mappings of the INM framework to the overall 4WARD architectural framework defined in the previous section are also well established. The Stratum and the SE are closely linked and this gives the potential to have self managing strata. The algorithms developed inside INM for anomaly detection are key information providers to the Knowledge Stratum which in turn feeds into the Governance Stratum for required actions which can be realized by the INM self-adaptation schemes.

INM, which is discussed in detail in [4], moves away from the traditional approach and deploys collaborating management logic across the network.

3.3 Network virtualization

Virtualisation has the potential to resolve the “deployment stalemate” observed in today’s Internet; it has therefore gained sufficient momentum as one of the key paradigms for future networking. The main use of network virtualisation within 4WARD is to permit the rapid deployment of new networks and architectures and to enable the co-existence of heterogeneous network architectures over a common infrastructure. The internal view of virtualisation can be decoupled in two main tasks: resources virtualization and virtual network provisioning and management.

Within the 4WARD architecture, the substrate stratum would be the set of physical resources owned by the infrastructure providers and composed of virtualizable nodes and links. On the other hand, the virtualization stratum would be managed by the virtual network provider to permit the deployment of on demand ad-hoc virtual networks (Vnets). From an external perspective, the virtualization stratum would be composed by interconnected virtual nodes and virtual links. Virtual nodes are system entities capable of hosting netlets. On the other hand, virtual links are the abstraction of the communication channels interconnecting virtual nodes.

The virtualization stratum and the substrate stratum are related by the virtual resources SSP, which orchestrates the request of individual virtual resources to the substrate resources owner. The virtualization stratum offers a virtual network SSP to upper strata, providing a set of interconnected virtual nodes and virtual links as well as some management interfaces. From the horizontal interworking perspective, the Folding Point is defined as the virtual network SGP. The Folding Point is the means to interconnect virtual networks running potentially different architectures, while keeping security, stability and confidentiality into account. The Folding Point is a composed SGP, with functions at the substrate (physical connectivity), virtualization (inter-Vnet operation) and connected endpoints strata (gateway functions).

Virtualization of an inter-network architecture, such as the Internet, involves virtualization of a variety of individual resources. Virtualization of a particular resource type may put forward a specific set of technical challenges. Despite the diversity of techniques, virtualization techniques should comply with the high level architectural requirements such as isolation, privacy, and QoS support.

Virtualization of nodes is based on hardware resources partitioning and isolation and leverages the recent developments in operating system virtualization, taking into account fairness and global performance. Virtualization of links extends the concepts of resource sharing at the link level by defining a common architecture for link partitioning, aggregation and emulation. These virtualization concepts can be potentially extended to other resources such as servers, switches, control plane elements or even other 4WARD concepts such as the Generic Paths.

3.4 Generic Paths

New mechanisms for data transport face contradictory requirements: large flexibility vs. uniform interfaces to all transport entities and efficient reuse of functionality are required. This can be partially achieved by new protocols only in end systems, but in general, an approach how to structure protocols both at the edge and in the core, at various “layers”. For example, network management needs to identify, inside the network, data flows of different types; they should be able to give account of themselves (e.g., about their desired data rate) and obey a common set of commands.

To support such requirements, we focus on the data flow and its path as a core abstraction, along with a design process for a variety of path/flow behaviours. This process can incorporate new networking ideas; examples are network coding, spatial diversity cooperation, or multi-layer routing and is suitable for both end system and in-network implementation; the deployment is supported by the Netcell architecture.

The starting point for the 4WARD transport architecture was to find (1) a development model that can support reuse and flexibility, (2) a proper execution environment within a node (end system or router) with naming and addressing structure and a resolution scheme, and (3) the core functions and APIs necessary for a path, as generic as possible. Together, this is the core of the Generic Path architecture. It approaches issue (1) by using an object-oriented approach to define types of Generic Paths and to structure their interfaces; issue (2) by defining a set of concepts (namely, entity, endpoint, mediation point, compartment, hooks, and path [12]) that describe the execution environment of instances of such path types; and issue (3) by selecting which operations should be possible on such paths (e.g., joining, splicing, or multiplexing). The concept shares some commonalities with OpenFlow [7], but concentrates on real-world necessities rather than on experimental usage; it also goes beyond merely modifying switching tables. To incorporate new networking ideas, all the relevant flows in a network share crucial commonalities and provide a common set of APIs with which to manipulate these flows. An example how to exploit such commonalities is realized in 4WARD's "Cooperation & Coding Framework" [11], an entity that detects opportunities for turning on cooperation opportunities like network coding and can create the necessary path instances to setup a network coding butterfly (other cooperation options like spatial cooperative diversity fit in this concept as well).

Based on this mindset, it becomes possible to develop powerful, custom-tailored path types. An example are path types for a Network of Information (described next), where the download of documents and the updating of location/caching tables can be tightly integrated and can access topology information to choose, for a document of interest, topologically close caches. Another example would be a path type to support the exchange of management information for In-Network Management entities, e.g., by compressing monitoring information more and more the further it is away from its source.

3.5 Networks of Information

Today's networking is essentially about exchanging information between nodes. When accessing information, the request typically includes the host that the information shall be retrieved from, frequently in the form of a Uniform Resource Locator. This host-centric approach is often an obstacle for optimized transport of and easy access to information. Our approach to an information-centric architecture puts the information itself on the centre stage. We take existing proposals that separate the host identity from the locator one step further by introducing information objects as first order elements in the network. In addition to classical scenarios such as content distribution [5], our work also encompasses scenarios that have so far not been discussed in the research community, e.g. the notion of real-world object tracking under the aegis of an information-centric architecture [6].

For the envisaged Network of Information (NetInf), we have developed an information model that constitutes a versatile and widely applicable framework for representing information in a wide sense. A clear split between the information itself and the location it is stored at is introduced. This eliminates the need for overloading locators and avoids putting them in the role of being an identifier and a locator at the same time. The representation of the actual files containing the payload is called a *data object* whereas the higher semantic level can be expressed by *information objects* that group or aggregate information. We have conducted a detailed analysis of requirements with regard to naming these objects and have proposed a first version of a naming framework that is suitable for securely naming the wide array of possible objects [6]. Objects may have associated meta-data, also to support search operations which bridge the gap between human-readable search terms and the flat, cryptographic identifiers of the objects. In addition to manual mechanisms, the 4WARD self-management capabilities can be used to create and update bindings between the objects.

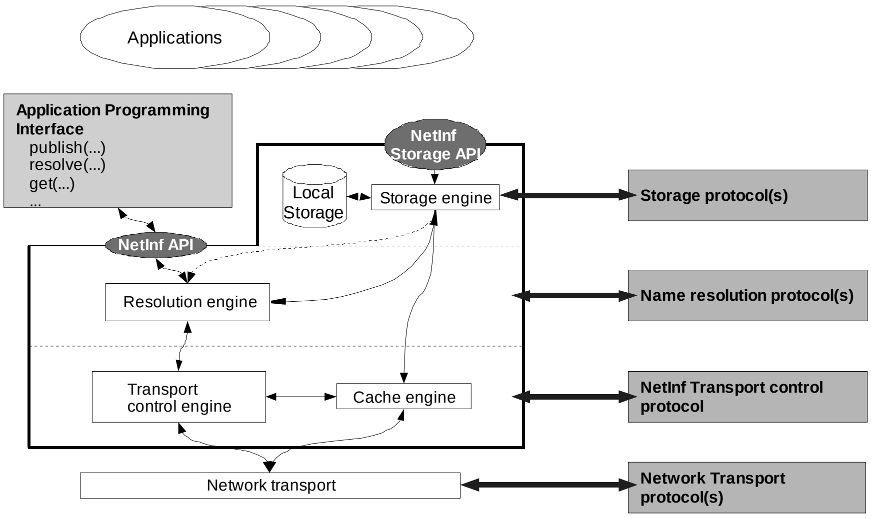

Figure 2: NetInf High-level Architecture.

The high-level architecture of a NetInf node is depicted in Figure 2. A uniform API exposed towards applications provides standard operations such as retrieving, publishing or updating information objects. Two routing schemes suitable for information-centric networking have been proposed, which both offer a scalable solution for handling the enormous amounts of bindings that need to be handled by the name resolution system. These Resolution Engines rely on the principle of multiple, concatenated resolution steps, where an update only needs to be made to the "closest" part of the dictionary. Complementing the mobility schemes offered by the underlying transport, these mechanisms also provide a means to not only handle the mobility of nodes and networks, but also information objects.

The NetInf Transport Control Engine is extremely flexible with regard to the transport mechanism that is utilized to transport the information objects or the requests. These transport mechanisms include, but do not mandate, the Generic Paths. Essentially, a set of adapted and optimized transport mechanisms applicable to information-centric networking are examples of specialized Generic Paths. The Transport Control Engine closely interacts with the Cache Engine which manages the caches that are used for short-term optimizations of data transport. The long-term memory of a NetInf system is provided by the Storage Engine. It uses the basic NetInf primitives to deliver and retrieve objects, while offering an advanced API that enables applications to manage the objects in the storage system.

The smooth deployment of NetInf will be supported by other 4WARD technologies such as virtual networks and the Netcell concept, which allow for an incremental and parallel roll-out of the architecture.

4 Conclusions and further work towards a system architecture

Considerable research effort is clearly necessary to address the challenges raised by the design of a Network of the Future. This effort is currently underway with many Future Internet activities across the world. The main thrusts of 4WARD, *a new architectural design*, the *information-centric paradigm, the generic path network virtualization and embedded self-management,* will provide candidate solutions, which, after careful evaluation, should be appropriately incorporated into one architectural framework – we foresee that no single architecture will be able to satisfy the networking needs of the future but that a common flexible and secure framework is needed instead.

Within 4WARD we have developed the stratum concept and evolved it towards the *Netcell,* which is based on the integration of concepts defined in Section 3. The Netcell defines the basic building block of a Future Internet, much like the cell in a body. The Nucleus of a Netcell provides the Netcell with its 'DNA', i.e. the principles, properties, and design patterns which controls and guides the functional composition and configuration of the Netcell itself. The Governance and Knowledge strata, see Section 3.1 above, need to be present in any Netcell. The Governance stratum continuously ensures that a proper set of horizontal strata (the VNet, Generic Paths, and Information Management strata in our example Netcell) are instantiated and configured. The services and control capabilities offered by a Netcell to users (other Netcells) as well as to an administrator, respectively, are provided through the Netcell Service Point (NSP). In its simplest form it will just be the aggregate of the SSPs of each participating strata forming the internal composition of the Netcell, but this service & control capability offering may be further limited depending on e.g. policies. The Netcell collaborates with other Netcells via the Netcell Gateway Point (NGP). This collaboration is basically under the control of the Governance Stratum. The Nucleus also determines with what other 'kinds' of Netcells a specific Netcell may collaborate with. Just as with the NSP, the NGP might just be the aggregate of the SGPs of each participating strata, but which may be further restricted due to policies. The NSP and NGP allows for a generalized approach to composition of Netcells, where the NSP can support so called service compositions [10] in order to create more complex or enhanced services. Through the NGP so called network compositions [9] can be performed in order to support internetworking and which may result in that Netcells become concatenated aggregated, merged, sliced etc, which would depend on the outcome of a network composition. Thus, the Netcell is itself a system on its own, but also a component which through composition is able to create new and/or more complex systems. As such, it is a system concept which is able to support in a highly modular and extensible way both agile networking, as well as agile and potentially complex business operations.

In the 4WARD project, we will continue to further evaluate, develop, and refine the architectural concepts, and at the end of the project we expect a final conclusion and description to be available.

5 References

[1] J. Day; Patterns in Network Architecture; Prentica Hall; 2008

[2] 4WARD project page at www.4ward-project.eu

[3] 4WARD project. Deliverable 2.1: "Technical Requirements"

[4] 4WARD project. Deliverable-4.2: "In-Network Management Concept"

[5] V. Jacobson, M. Mosko, D. Smetters, and J. J. Garcia-Luna-Aceves. Content-centric networking. Whitepaper, Palo Alto Research Center, January 2007.

[6] 4WARD project. Deliverable D-6.1: "First NetInf architecture description"

[7] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker and J. Turner: "OpenFlow: enabling innovation in campus networks," ACM SIGCOM, Vol. 38, No. 2, 2008.

[8] C. Dovrolis, "What would Darwin think about clean-slate architectures?" in ACM SIGCOMM Computer Communications Review, vol. 38, no. 1, pp. 29-34, Jan. 2008

[9] M. Johnsson, A. Schieder, and R. Hancock. "Final System Description", Public Deliverable D18-A.4, Ambient Networks project, Jan. 2008.

[10] H. Bannazadeh, A Leon-Garcia, "On the Emergence of an Application-Oriented Network Architecture", IEEE International Conference on Service-Oriented Computing and Applications, 2007. SOCA apos;07, 19-20 June 2007, ISBN 0-7695-2861-9, pages 47 – 54.

[11] T. Biermann, Z. Polgar, and H. Karl, "Cooperation and Coding Framework", Proc. International Workshop on the Network of the Future (Future-Net), colocated with IEEE ICC, June 2009.

[12] 4WARD project Deliverable-5.2.0: "Description of Generic Path Mechanisms"