IPv6 and Extended IPv6 (IPv6++) Features That Enable Autonomic Network Setup and Operation

Ranganai Chaparadza[1], Razvan Petre[1], Arun Prakash[1], Felicián Németh[2], Sławomir Kukliński[3], and Alexej Starschenko[1]

[1] Fraunhofer FOKUS, Berlin, Germany
{ranganai.chaparadza,razvan.petre,arun.prakash,
alexej.starschenko}@fokus.fraunhofer.de
[2] Budapest University of Technology and Economics, Budapest, Hungary
nemethf@tmit.bme.hu
[3] Warsaw University of Technology, Warsaw, Poland
kuklinski@tele.pw.edu.pl

Abstract. In this paper we present an insight on the IPv6 features and a few examples of propositions for Extensions to IPv6 protocols, which enable autonomic network set-up and operation. The concept of autonomicity-realized through control-loop structures embedded within node/device architectures and the overall network architecture as a whole is an enabler for advanced self-manageability of network devices and the network as a whole. GANA Model for Autonomic networking introduces *autonomic manager components* at various levels of abstraction of functionality within device architectures and the overall network architecture, which are capable of performing autonomic management and control of their associated Managed-Entities (MEs) e.g. protocols, as well as cooperating with each other in driving the self-managing features of the Network(s). MEs are started, configured, constantly monitored and dynamically regulated by the autonomic managers towards optimal and reliable network services. This amounts to what we call autonomic setup and operation of the network. We present how to achieve this, and also present the features that IPv6 protocols exhibit, that are fundamental to designing and building self-configuring, self-optimizing and self-healing networks i.e. IPv6 based autonomic networks.

Keywords: IPv6, Evolution of the current Internet towards Self-Managing Future Internet.

1 Introduction

The main benefits of the self-management technology in systems and networks, from the operator's perspective are: to minimize operator involvement and OPEX in the deployment, provisioning and maintenance of the network, and increasing network reliability (self-adaptation and reconfiguration on the fly).

R. Szabó et al. (Eds.): AccessNets 2010, LNICST 63, pp. 198–213, 2011.

The FP7 EFIPSANS project [1] introduced a standardizable evolvable Architectural Reference Model for Autonomic Networking and Self-Management dubbed the **G**eneric **A**utonomic **N**etwork **A**rchitecture (**GANA**) [2,3]. The GANA Model defines fundamental building blocks that should be considered when designing devices of a network that is autonomic/self-managing. GANA, presented in brief, in the next section, is a holistic Architectural Reference Model for Autonomic Network Engineering and Self-Management that serves the following purposes: (1) To answer the question of how Self-Management/Autonomicity can be introduced into the fundamental architecture of Future Internet devices (GANA-conformant devices), (2) To then instantiate GANA with autonomic management and control of Protocols and Mechanisms (e.g. IPv6 protocols, because core GANA concepts are protocol agnostic), (3) To use GANA as a guide to examining and exploiting the strengths and features of IPv6 protocol, e.g.in order to have the *big picture* on where Extensions to IPv6 protocols (IPv6++) can be introduced and for what purposes.

This approach to designing the self-managing Future Internet has lead to the Propositions for a number of Extensions to IPv6 Protocols (IPv6++) being proposed by EFIPSANS. For more information on the kind of extensions to IPv6 that are necessitated by a GANA compliant network, we refer the reader to [2,3,4]. In this paper, we present a few selected IPv6 Extensions required for autonomic networking.

2 Fundamentals of Autonomic Networking and Self-management

The concept of autonomicity-realized through control-loop structures embedded within node/device architectures and the overall network architecture as a whole is an enabler for advanced self-manageability of network devices and the network as a whole. The GANA model introduces *autonomic manager components* at 4 various levels of abstraction of functionality within device architectures and the overall network architecture, which are capable of performing autonomic management and control of their associated Managed-Entities (MEs) e.g. protocols, as well as co-operating with each other in driving the self-managing features of the Network(s). MEs are started, configured, constantly monitored and dynamically regulated by the autonomic managers towards optimal and reliable network services. The four levels are *protocol-level*, *abstracted-functions-level*, *node-level* and *network-level* [2]. A central concept of GANA is that of an autonomic Decision-Making-Element (**DME** or simply **DE** in short-for Decision Element). A Decision Element (DE) implements the logic that drives a control-loop over the *management interfaces* of its assigned Managed Entities (MEs). Therefore, in GANA, self-* functionalities such as self-configuration, self-healing, self-optimization, etc, are functionalities implemented by a Decision Element(s). The fundamental principles of the setup and operation of an autonomic network can be described as three cascaded phases of some automated behaviors of nodes/devices being connected together to form an autonomic network, namely:

[**Phase-1**] **-** Boot-up and Bootstrap Phase for each initializing node/device; [**Phase-2**] **-** Auto-Configuration Phase for each node/device and the network as a whole; [**Phase-3**] **-** Operation and Self-Adaptation Phase for each node/device and the network as a whole, i.e., adaptation to challenges and adverse conditions and policy changes by the human.

The following automated behaviors of node/devices and the network (realized as autonomic behaviors orchestrated or triggered by autonomic managers) apply to some of the specific phases (from the three described above). **Auto-Discovery** (Network-Layer-Services Discovery, Service/Application-Layer-Services Discovery) - Applies to [**Phase-1**], and some behaviors related to Auto-Discovery for more advanced service provisioning requirements beyond the minimal required at bootup/bootsrap time, may still be attempted during the operation and self-adaptation time of a node/device or network; **Auto-Configuration/ Self-Configuration** (in the Service-Layer and Network-Layer) - Applies to [**Phase-2**]; **Self-Diagnosing and Self-Healing, Self-Optimization, other Self-* functions** - Applies to [**Phase-3**].

3 Need for a Information and Knowledge Sharing System

In an autonomic network, different network entities need to collaborate in order to fulfill the global goals of the network. For an efficient collaboration, data (e.g. resources and capabilities description, configuration data, events or alarms), information and knowledge must be gathered, elaborated and shared between the different network entities. Hence, a powerful system for exchanging information and knowledge must be in place. The information and knowledge exchange system enables more advanced autonomic and cognitive functions like self-configuration, auto-discovery, self-adaptation, self-optimization or other self-* functionalities. Since the information and knowledge exchange system is a fundamental requirement for achieving autonomic and self-management functionalities, it must scale, it must be fault-tolerant and provide a good accessibility, and it must be secure.

Figure 1 highlights the role of the information and knowledge exchange system inside a GANA network. For special type of information and knowledge like time critical information, very large pieces of information or fast changing information, other protocols and/or mechanism must be in place. All the other types of information and knowledge are shared through the exchange system. In the same time, any node may have a local repository for storing information and knowledge. The key requirements and goals that must be considered when designing such a system are:

- It must support a large number of resources in large-scale autonomic networks. Thus, the system should scale to a large numbers of resources spread throughout a wide area network across different administrative domains. To achieve this goal, the system must be distributed ideally as a managed overlay network of information/knowledge servers (e.g. one possibility would be to rely on distributed hash tables - DHT). A resource is understood as a

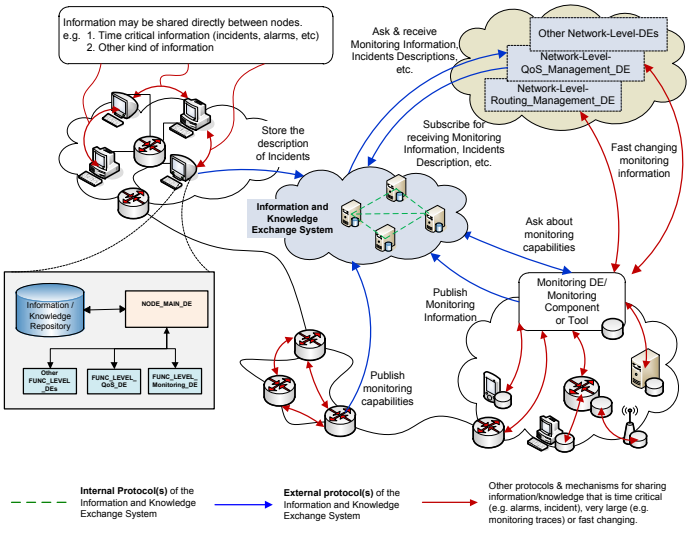

Fig. 1. The Role of Information and Knowledge Sharing exchange system in a GANA Conformant Network

device or services offered by devices, but also as data in the form of configuration data, network policies, incidents descriptions, monitoring data, network knowledge, etc. The system must also provide scalability to support a large number of updates and data-retrieval requests.

- It must be highly available, as it plays an important role in the processes of Auto-Discovery and Auto-Configuration and other Self-* functionalities. To achieve this goal the information and knowledge stored inside the system must be replicated and must be always available, even in the case of failures of several servers forming the distributed system.
- The information and knowledge exchange system must offer the possibility to update information that was previously published. It must provide very good update times for replications for all data types.
- It must support complex queries. Partial queries should be supported (queries that contain only a subset of the attributes originally advertised by resources, considering the other attributes as wildcards). Each query might also have a scope (global, local, n-hops away, etc).
- It must provide a powerful subscription mechanism for receiving information of interest.
- It must provide well-designed interfaces to allow efficient data input and retrieval. These interfaces may be summarized as the system's external protocols. The messages to and from the Information Exchange System should be optimized so as not to contribute to congestion in the network.
- It must provide well-designed communication protocols and primitives to be used by servers that are part of the overlay network, to build and maintain the overlay network. These protocols and primitives must facilitate the efficient distribution and propagation of information/knowledge across the

overlay network and also must support advanced services like replication, clustering or security. They may be summarized as the internal protocols of the system.
- It must provide well-designed communication protocols and primitives to be used by any network entity willing to access the services offered by the system. These communication protocols and primitives may be summarized as the external protocols of the system.
- It must be easily extendable to allow the inclusion of future functionality.
- Since the goal of GANA is to reduce the complexity in management network, the Information and Knowledge Exchange System must be optimized for simplicity of installation and maintenance.

The instantiation of such a system is the ONIX system. ONIX stands for **O**verlay **N**etwork for **I**nformation e**X**change and it is the EFIPSANS proposed solution for a scalable, fault-tolerant and secured information and knowledge exchange system in IPv6 networks, but not limited to it. The architecture of the ONIX system is presented in Figure 2.

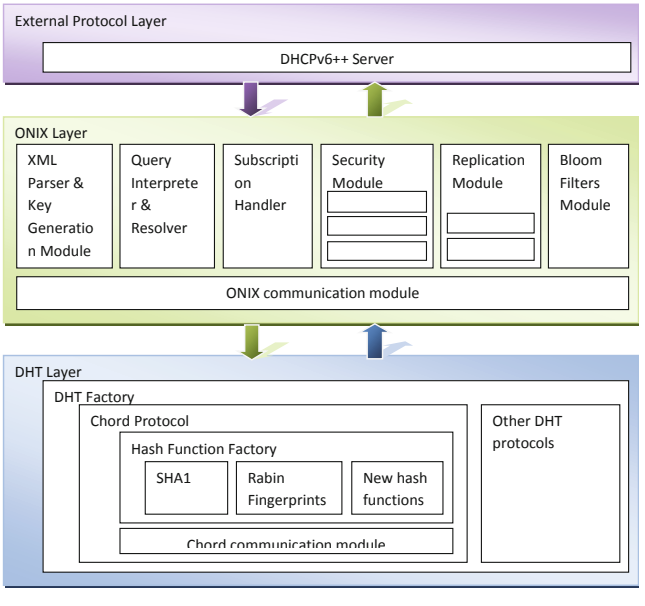

Fig. 2. ONIX System Architecture

ONIX system is built on top of a DHT. In the current implementation Chord [5] protocol is used, but any DHT protocol that expose the functions: put(key, value) and get(key) may be used as the underlying DHT protocol. Example of other possible DHTs include [6,7,8]. Moreover, if the DHT protocol permits, ONIX system provides a mechanism to easily switch between different hash functions, like SHA1 or Rabin Fingerprints [9]. Information and knowledge published

to the system is described using XML, and the query language is based on Xpath. Keys and data, information and knowledge are replicated inside the system, the component responsible for handling the replication process as well as for keeping the replicas synchronized is the Replication Module. Bloom Filters [10] are used to reduce the traffic generated in the system for resolving complex queries. The ONIX internal protocol is an extended version of the Chord protocol. As the external protocol, the system uses an extended version of DHCPv6 (DHCPv6++), designed in EFIPSANS project. A short summary of the services offered by the ONIX system includes:

- Information and Knowledge storage and retrieval : (a) Push & Pull models supported, (b) Different classes of information, (c) Add, remove or replace operation supported, (d) Information is described using (XML).
- Information Query: by supporting a query language capable of expressing complex queries (partial queries, scoped queries, etc) (based on XPath).
- Information Dissemination: Upon request, periodically or event triggered: (a) Normal Subscription, (b) On-behalf Subscription, (c) Publish & Disseminate
- Security: Authentication, Authorization, Trust, Confidentiality, Integrity, Non-repudiation, Privacy, Tracking of activities taken and originators of each input to the system for accountability and auditing.
- Reliability, fault-tolerance and accessibility

4 Auto-discovery as an Enabler for Self-configuration and Self-adaptation

A self-managing network needs a way to know the entities composing the network in order to employ self-configuration and self-adaptation mechanisms to configure and operate the nodes in the network. In the context of GANA , the Auto-Discovery functionality consists of Self-Description, Self-Advertisement, and support for Solicitation of *Capabilities* at the node level and Topology-Discovery. In the context of GANA, we define these as follows:

Self-description is defined as the ability of a functional entity to *describe* itself. This includes the description of its *Capabilities* such as software and hardware specifications, available services and tools, supported protocols, node interface information including its current role and a list of potential roles it can play in the network. The self-description mechanism of a node results in the generation of the *GANA Capability Description Model* [11] of the node.

Self-Advertisement of Capabilities is the process by which a functional entity spontaneously disseminates the generated *GANA Capability Description Model* to other functional entities either inside a node or in the network subject to security policies. The dissemination may be carnied out through an information and knowledge sharing repositories such as ONIX.

Support for solicitation for Capabilities is ability of a functional entity to respond to requests for its Capability Description by initiating its self-description and self-advertisement functions. This is vital for the self-organization functionality of a network.

Topology-Discovery is the ability of the functional entity to automatically discover the topology of its network without any manual assistance. Topology-discovery is essential to detect the presence of new nodes, the absence of nodes due to failures and changing network conditions. In the context of GANA, the topology-discovery function can be performed by the NET_LEVEL_RM_DE. Each node publishes its neighbor information, obtained by using IPv6's Neighbor-Discovery (ND) protocol, to the NET_LEVEL_RM_DE in order to facilitate the computation of the network topology.

In GANA, the auto-discovery mechanism is initiated by the NODE_MAIN_DE of a node. The NODE_MAIN_DE generates the *Capability Description* for the node by triggering the iterative self-description process as shown in Figure 3. The NODE_MAIN_DE requests for the *Capabilities* of its underlying Function-Level DEs. These DEs in turn request for the *Capabilities* of the Protocol-Level DEs and MEs. Thus the *Capabilities* of individual DEs and MEs are obtained in a recursive manner. This completes the self-description process of the node. The aggregated Capabilities of the node are then published to on-link neighbors and to ONIX for a network wide dissemination. This completes the self-advertisement process of the node. The self-description and self-advertisement functions are repeated every time the *Capabilities* of the node changes due to failures, software updates and changing networking conditions.

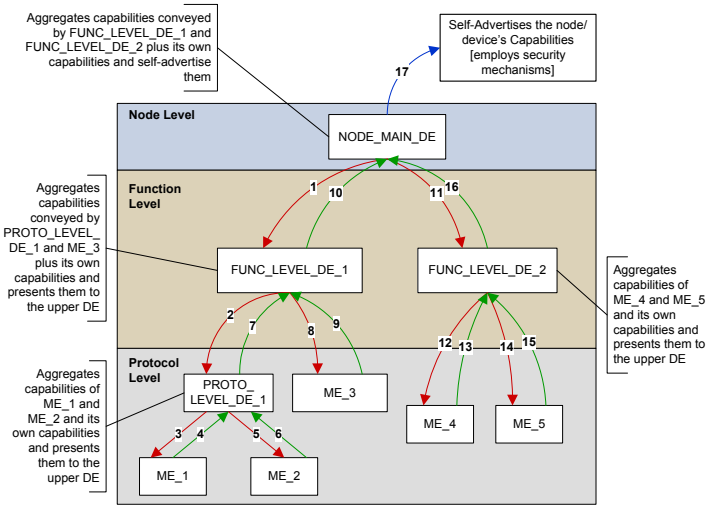

Fig. 3. Self-Description and Self-Advertisment of GANA Capability Description

For the Topology-Discovery mechanism employed by the NET_LEVEL_RM_DE, the neighbor information of each node in the network is required. The neighbor information of each node is used for constructing the network topology graph. Each node publishes its neighbor information, obtained by using IPv6's Neighbor-Discovery (ND) protocol, to the NET_LEVEL_RM_DE. The neighbor information is provided by the NODE_MAIN_DE by publishing and updating a list of

its on-link nodes on each interface to the NET LEVEL RM DE. Every time the *neighborhood* of a node changes either due to failures or due to the bootstrapping of new nodes, or due to dynamic network conditions, the NODE_MAIN_DE updates the NET_LEVEL_RM_DE with new neighbor information. This facilitates the topology-discovery function to be in sync with the changing network topology. For a detailed sketch on the *GANA Capability Description* and the algorithms describing the behavior of the auto-discovery mechanism in GANA, the reader is directed to [11]. The information obtained through the auto-discovery mechanisms enable Network-Level DEs to compute node configurations and decisions for various self-adaptation mechanisms in the network.

Auto-Configuration in a GANA conformant network is achieved through the use of the *GANA Network Profile* (GANA NETPROF) [11]. The GANA NETPROF is composed of a NETPROF, GANA Configurations Options Map (MAP) and several vendor specific configuration files. The NETPROF can be considered as an entity that provides a structural and monolithic framework for defining *policies*, *objectives*, *high-level DE configurations* and for the network and its nodes, along with hooks for adding vendor specific configurations for the nodes. Thus in a NETPROF, the policies, objectives and high-level configurations are categorized in terms of the various node *roles* planned in the network, rather than the actual nodes available in the network. This allows a vendor agnostic implementation of the DEs. As MEs are vendor specific by the nature of their design and implementation, all vendor specific configuration options are delegated to the vendor specific files of the GANA NETPROF. The configurations provided to a node, the *GANA Node Configurations* (NODECONF), are specific to the vendor type of the node, and are generated from the NETPROF.

As a node boots up in a network, the NODE_MAIN_DE bootstraps its interfaces and initiates the self-description process as shown in Figure 3. The neighbor-information is also computed side-by-side. The aggregated *Capabilities Description* and neighbor-information are published to ONIX and on-link neighbors. ONIX ensures the dissemination of the information to the Network-Level DEs. The NET_LEVEL_RM_DE uses the neighbor-information for the computation of the network topology-graph (topology-discovery). If OSPF routing is an objective of the network, the obtained network topology-graph is used for partitioning the network into OSPF areas. Thus new areas are formed when the number of routers in an area conflicts with a policy delineating the threshold number of routers for an OSPF area. Existing areas are merged when node failures occur and the number of nodes in a given area falls below the minimum number of routers in an area required. Thus to successfully employ OSPF routing, the network self-adapts to changing network conditions, by partitioning and merging OSPF areas, a behavior executed by the NET_LEVEL_RM_DE.

In the context of OSPF routing, the node roles are classified as follows: Core Router (CR), Area Border Router (ABR) and Autonomous System Border Router (ASBR). The *Capability Description* of a node provide information regarding the vendor, software and hardware attributes and the current role of the node in the network. These, along with the partitioning information is used

by the NET_LEVEL_RM_DE for the computation of the NODECONF. The NETPROF is searched for the appropriate node role sub-profile and the vendor specific configuration files in order to generate the NODECONF. The exact nature of the algorithm used for the NODECONF generation is described in [11]. Once the NODECONF is generated, it is disseminated to the respective nodes through ONIX. The NODE_MAIN_DE receives the NODECONF and uses it to self-configure its DEs and MEs to reflect the dynamic objectives of the network.

5 Autonomic Routing in Wired Network Environment

The Routing Functionality (Function) of nodes in an IPv6 based fixed network and the network as whole can be made autonomic by making diverse Routing Schemes, Policies and Routing Protocol Parameters employed and altered based on network-objectives, changes to the network's context and the dynamic network views in terms of events, topology changes, etc. Figure 4 depicts how the routing behavior of a node/device and the network as a whole can be made autonomic.

Two types of Control-Loops are required for managing/controlling the routing behavior. The first type is a node-local control loop that consists of a Function-Level Routing-Management DE embedded inside an autonomic routing node e.g. a router. The local Function-Level Routing-Management DE is meant to process only that kind of information that is required to enable the node to react autonomically and autonomously(according to some goals) by adjusting or changing the behavior of the individual Routing protocols and mechanisms required to be running on the node. The Function-Level Routing-Management DE reacts to *views*, such as *events or incidents* exposed by its Managed Entities (MEs) i.e. the Routing protocols and mechanisms. Therefore, the Routing-Management DE implements self-configuration and dynamic reconfiguration features specific to the routing functionality of the routing node. It is important to note that due to scalability, overhead and complexity problems that arise with attempting to make a Routing-Management DE of a node process huge information/data for the control loop, a logically centralized Decision Element(s), may be required, in order to relieve the burden. In such a case, a network-wide slower Control Loop is required in addition to the faster node-local control-loop (with both types of loops working together in controlling/managing the routing behavior in an autonomic way). In [12] an instantiation case of GANA for autonomic management and control of IPv6 routing protocols and mechanisms, as discussed in this section, is illustrated and elaborated. More details on the inter-working of the 2 DEs (control loops) and the aspects related to cognition can be found in [2]. In the [2], related work on how this framework is applied to Auto-Discovery and Auto-Configuration of Routers in an Autonomic Network is presented.

Next we show using the example of Open Shortest Path First (OSPFv3) protocol how we examined the Features in IPv6 protocols that are fundamental to designing and building self-configuring, self-optimizing and self-healing networks i.e. IPv6 based autonomic networks. Each paragraph below corresponds to raw

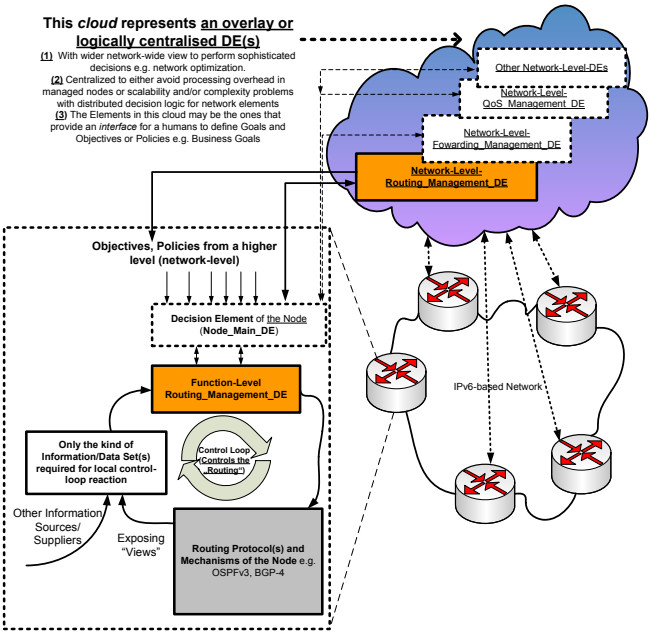

Fig. 4. Autonomicity as a feature in Routing Functionality in a IPv6 based Network

in our questionnaire template. Additionally, Table 1 enumerates the basic features of OSPFv3 and shows how and where they fit into the GANA framework. Each feature is examined from three aspects detailed in the header row.

Summary of the usage of the Protocol and any of its exploitable features. OSPFv3 is a link-state routing protocol that is associated with the control plane of today's IP networks. The protocol comes with a management interface that can support the automation of (re)-configuring the protocol's behavior according to the goals set for the routing behavior of the network for which OSPFv3 is meant to fulfill. CLI types of (re)-configuration by a human are normally supported by most implementations of OSPFv3. For the management of OSPFv3 by traditional centralized NMS-type of approaches to network management, the OSPFv3 MIB can be used to perform the (re)-configuration of the protocol. Management of the protocol from within the node running an OSPFv3 instance may require some different approach to CLI or MIB types of interfaces. OSPFv3 implements a simple control loop that implements a self-adaptation mechanism to link failures as described later in this table.

Mapping the IPv6 protocol to the GANA Functional Planes. OSPFv3 belongs to the *Control Plane*, a sub-plane of *Dissemination Plane* of GANA. Information such as Routes and link state information is disseminated among OSPFv3 supporting nodes. Because OSPv3 implements some control loop, we can loosely associate it with some kind of Decision Element intrinsic to the protocol itself by design. Therefore, we can consider such kind of a protocol intrinsic DE as belonging to the Decision Plane of GANA. Because there is no separation between the

Table 1. Exploitable OSPFv3 Features

Protocol Feature that can be exploited	**1. Any Decision Element (DE) that can use the Protocol Feature(s) [Case3].** ***use*** **means getting information supplied by or via the protocol feature to aid the decision making process of the DE OR managing the use of the available feature for some purpose/goal.** **2. Network Environment(s)** **3. Any other Self-* Behavior that can benefit from the feature (if applicable)**
OSPF Hello Parameter Configuration	1. The DE at the Network-level responsible for self-configuration/self-adaptation of protocol behavior could (re)adjust OSPF Hello timers to rate-limit Hello traffic.
	2. Fixed or slowly changing wired or wireless network environments.
	3. Self-configuration and self-optimization.
OSPF Interface Configuration	1. The DE at the Network-level responsible for self-configuration MUST provide a routing profile (IP addresses, identity, link cost, etc.) for OSPF to bootstrap its operation on interfaces where needed.
	2. Any network environment.
Area Border Router (ABR) Support	1,2 N/A
OSPF Area Support	1. The DE at the Network-level responsible for self-configuration/self-organization could restructure the areas in order to limit/optimize LS flooding traffic. For instance, stub (sub)-networks could be separated into distinct areas.
	2. Fixed, relatively larger, heterogeneous wired topologies, where the amount of LSA traffic could be substantial.
Route Redistribution	1. The DE at the Network-level responsible for self-configuration might readjust route import/export from/to external, inter-domain routing protocols to holistically optimize the interaction of intra- and inter-domain routing (e.g., to avoid hot-potato routing).
	2. Fixed, relatively larger, heterogeneous wired topologies.
Type 1 and Type 2 External Routing Support	1,2 see above
Virtual Link Support	1. In consequence of the reorganization of the area-structure, readjustment of virtual links might become necessary to ensure the consistency of the Backbone area (area 0.0.0.0). This is the task of the Network-level Routing Management DE.
	2. Fixed, relatively larger, heterogeneous wired topologies
Unknown LSA	1,2 N/A
OSPF Management Information Base (MIB)	1,2 N/A
Traffic Engineering Support	1. Network-level Self-optimization functionality might use resource availability information disseminated in extended TE-LSAs to make informed Traffic Engineering decisions.
	2. Any network environment
	3. Monitoring
Non-broadcast Multi-Access (NBMA) Support	1,2 N/A

decision logic i.e. a DE that implements the control loop intrinsic to OSPFv3 and the rest of the functions of the OSPFv3 that can be considered as regulated (managed) by such a virtual DE, the protocol itself (as a single module at implementation and run-time), as a whole, can be considered as belonging to the Decision Plane of GANA. This means that OSPFv3 is *neutral* to both the Decision Plane and Dissemination Plane of GANA. Note: Some *autonomic protocol* can be designed in such a modular way that it clearly has a *distinct* separation between its protocol-intrinsic DE and the regulated(managed) functions of the protocol that are managed/regulated by the associated protocol-intrinsic DE.

Any Self-* Behavior that can be considered as a feature intrinsic within the protocol. OSPv3 as a link-state routing protocol, supports *Self-Adaptation* by performing failure detection and adaptation to the underlying network topology and link or node failures. After link weights have been configured (typically a manual process), the routing protocol will discover the network topology, disseminate routing information and set up consistent forwarding tables. Additionally, OSPFv3 is capable to adapt to failures: it detects link failures through a variety of methods, such as repeated failures to receive packet acknowledgements. Following failure detection it will reroute, eventually converging on a new valid path. Some forms of *Auto-Discovery*, *Self-Description*, *Self-Advertisement* and *Self-Organization* functionality can also be identified in the operation of OSPFv3.

Any other Self-* Behavior that can benefit from using the protocol in general. *Self-Optimization* functionality can be achieved by optimizing shortest paths. This involves extending the OSPFv3 LS flooding protocol to convey resource (bandwidth) availability at network links and incorporating OSPFv3 into a control loop that fine-tunes the link costs with respect to monitoring data and actual user demands. This *Self-Optimization* functionality can be extended to involve multipath routing (OSPF-OMP). Additionally, proposals exist to facilitate for fast OSPFv3 *Self-Healing* (IPFRR).

Any Decision Element (DE) that can manage the protocol [Case-1] OR is intrinsic to the protocol [Case-2]. The Routing-Management DE of an autonomic node, operating on the *abstracted networking functions level* of GANA-HCLs framework, that manages all routing protocols and mechanisms on the node, is the DE considered as the *autonomic manager* of OSPFv3 and other routing protocols and mechanisms of the node (collectively). By design, OSPFv3 does not have a *distinct* DE intrinsic within the protocol per se.

6 Autonomicity for the Data Plane

The Data Plane and the Forwarding functionality of nodes and the network as a whole in an IPv6 based network can be made autonomic by making Diverse Forwarding schemes and GANA Data-Plane parameters such FIBs, ACLs, packet filters, etc, employed and changed based on network-objectives, changing network context and the dynamic network views in terms of events, topology changes, etc. Like *Autonomic Routing*, two types of Control Loops are required for managing/controlling the data plane and the forwarding behavior. The first type is

a node-local control loop (the faster control loop) that consists of a *DataPlane-and-Forwarding-Management DE* (FUNC_LEVEL_DP_FWD_M_DE) embedded inside an autonomic node e.g. a router. The FUNC_LEVEL_DP_FWD_M_DE of a node is meant to process only that kind of information that is required to enable the node to react by adjusting or changing the behavior of the Data plane protocols, which include IPv6 Forwarding, Layer2.5-Forwarding, Layer2-Forwarding, Layer3-Switching, Layer2-Switching, etc, supported by the node. The FUNC_LEVEL_DP_FWD_M_DE reacts to *views*, such as *events or incidents* exposed by its MEs - the GANA Data-Plane protocols and mechanisms. The DE thus implements the self-configuration and dynamic reconfiguration features specific to the Data Plane and the forwarding functionality of the autonomic node. The node-scoped FUNC_LEVEL_DP_FWD_M_DE also relays *views* such as *events or incidents* to the *Network-Level DataPlane-and-Forwarding-Management DE* i.e. network-level control loop (the slower control loop) for further reasoning (in case, wider global knowledge is required in addressing the problems affecting the Data Plane and the forwarding behavior).

7 IPv6 in Autonomic Wireless Networks

Wireless ad-hoc Mesh Networks (WMNs) can serve as community networks, temporary networks for event handling or as sensor networks. WMNs are usually built using 802.11 devices due to their popularity, very low cost, royalty-free deployment and high link throughput. It is worth to emphasize that there are about 120 routing protocols developed for WMNs, 30 of them are expired IETF drafts, 2 are active IETF drafts, and 4 are IETF RFCs. Unfortunately, many of these protocols are well suited for a specific scenario only (for example for highly mobile or static networks, for sparse or dense networks etc.) and as a result it is very hard, if not impossible to select a proper protocol for a specific case.

In most WMNs throughput degrades exponentially with the number of wireless hops as a result of the usage of single radio channel, which is the case with most of the deployed or tested networks. The use of multiple radio channels may efficiently improve the overall network performance whereas the cost of adding extra radio interfaces to 802.11 nodes is negligible. In order to use multiple radio channels an algorithm and a protocol for channels allocation is needed. Another mechanism that may improve the performance of WMNs is the use of the multi-path approach, which may improve transfer reliability and increase the end-to-end throughput. In case of proactive routing and especially in case of multi-path routing, of great importance is the selection of the best path or paths for data forwarding. The path quality typically is evaluated by routing metrics. In WMNs a plethora of routing metrics are used, some of them especially designed for WMNs (namely AirTime, ETX and ETT [13]). It has been shown that proper metric selection have an important impact on the network performance [14]. In many WMNs routing protocols the hop-count metric is

used, which selects the shortest path in terms of number of hops, but is not load or interference aware. So, in autonomic WMNs more sophisticated metrics have to be used. Unfortunately, in the existing protocols, the metric is an integral part of the routing protocol and there is no easy possibility for their change. Another, generic problem of WMNs lies in ignoring by many approaches the information about the physical layer. The usage of the information about the physical and link layer characteristics (the cross-layer approach) has not only a positive impact on the routing (the avoidance of unreliable and low throughput paths), but also on radio channel selection and configuration.

There is an ongoing work on the IEEE 802.11s standard, which standardize WMNs, unfortunately this standard does not support multi-interface nodes, multi-path routing, energy-efficient operations nor advanced auto-configuration schemes. The management of 802.11s networks is centralized, thus not well suited for autonomic networks with dynamic topologies; there is no support for real-time distributed management operations that can be used for the optimization of the network behavior, according to user requirements and/or environmental changes. Therefore, a new approach to WMNs is required. This approach should cope with proper, dynamic routing protocol selection and appropriate tuning of the routing protocol used for the specific use-case, according to users or application preferences, network density, energy saving importance, nodes mobility, etc. This approach should enable a dynamic selection of routing metrics, and the dynamic choice of the path for data forwarding according to path quality, expressed in terms of: load, path length (number of hop-counts), path SNR, etc. This new approach should also provide support for cross-layer operations (physical and link layer monitoring and control), radio channel management and multi-path routing. All the above mentioned mechanisms if implemented would increase the reliability and the performance of the WMN, and bring about *autonomic WMNs*, which are able to adapt to the environment in a much more flexible way than is achieved by the use of current solutions.

This holistic approach requires monitoring and control of the nodes' behavior in near real-time. In order to implement such kind of autonomic management, the GANA Model can be used. As a result, the routing protocol can be dynamically fine tuned, multiple routing protocols can be employed on-demand and the multiple radio interfaces can be handled. In order to make all the above mentioned operations possible, we developed a Wireless and Autonomic Routing Framework (WARF). WARF has a modular structure, in which functions like forwarding, routing, radio channel management and policy control are separated. That way it is possible to modify one of the components leaving the other ones untouched. We use the IPv6 protocol with some Extensions (so called WARF Extensions to IPv6) to carry all WARF control messages. WARF control messages are based on the commonalities of the existing WMNs routing protocols. The uniform form of routing messages enable multiprotocol approach - the control messages have the same format for every routing protocol, but of course every protocol handles them differently. The WARF components enable fine tuning

of protocol parameters according to current needs. An outstanding feature of WARF is integration with routing, the channel allocation, and monitoring part, which is able to cope with multiple radio interfaces and control the physical layer properties (channel number, transmitted power etc.). IPv6 provides the flexibility to implement the mentioned functions, but also makes it possible to increase the control plane efficiency - the WARF messages can be piggybacked with user data packets, using the IPv6 Extension Header mechanism, in that way reducing the number of MAC requests. We distinguish four WARF basic components: Resource Maintenance Route Maintenance, Route Representation, Data Forwarding and Policy Control component (WARF management). A more detailed description of WARF can be found in [15]. The WARF Extension Header after validation in real testbed is intended for submitting as an IETF Internet Draft proposal.

8 Conclusion

In this paper, we presented an insight on the IPv6 features and a few examples of propositions for Extensions to IPv6 protocols (IPv6++), which enable autonomic network set-up and operation. The GANA Model presents a framework based upon which architects can reason about an autonomic network in its holistic sense. The work we presented here illustrates how the GANA Model can be instantiated for autonomic management and control of IPv6 protocols and lower layer transport and control-plane protocols and mechanisms to achieve an IPv6 based autonomic/self-managing network, capable of auto-discovering, auto-configuring (self-configuring) and self-adapting its resources to challenges such as adverse conditions, incidents, as well as policy changes received from the human operator. We presented the enablers i.e. enabling methods and mechanisms to designing an autonomic/self-managing network, including a distributed information/knowledge sharing system. What we also presented is a method to view the GANA as a guide to examining and exploiting the strengths and features of IPv6 protocols in order to have the *big picture* on where Extensions to IPv6 protocols (IPv6++) can be introduced and for what purposes. In that regard, such approach to designing the self-managing Future Internet has lead to a number of Extensions to IPv6 Protocols (IPv6++) being proposed by the FP7 EFIPSANS project. For more information on the kind of extensions to IPv6 that are necessitated by a GANA compliant network, we refer to the project deliverables available soon on the project's website. We presented selected examples of the proposed Extensions to IPv6, and illustrate an IPv6 based autonomic network, and associated architecture in which routers perform auto-discovery functions and auto-configuration (self-configuration).

Acknowledgment. This work is partially supported by EC FP7 EFIPSANS project (INFSO-ICT-215549) [1].

References

1. EC FP7-IP EFIPSANS Project (2008-2010), `www.efipsans.org` INFSO-ICT-215549
2. Chaparadza, R.: Requirements for a Generic Autonomic Network Architecture (GANA), suitable for Standardizable Autonomic Behavior Specifications for Diverse Networking Environments. In: International Engineering Consortium (IEC), Annual Review of Communications, vol. 61 (2008)
3. Chaparadza, R., et al.: Creating a viable Evolution Path towards Self-Managing Future Internet via a Standardizable Reference Model for Autonomic Network Engineering. In: Towards the Future Internet - A European Research Perspective, pp. 313–324. IOS Press, Amsterdam (2009)
4. Chaparadza, R.: Evolution of the current IPv6 towards IPv6++ (IPv6 with Autonomic Flavours. In: International Engineering Consortium (IEC) Annual Review of Communications, vol. 60 (December 2008)
5. Stoica, I., et al.: Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications. IEEE/ACM Transactions on Networking 11(1), 17–32 (2003)
6. Zhao, B.Y., et al.: Tapestry: A Resilient Global-Scale Overlay for Service Deployment. IEEE Journal on Selected Areas in Communications 22(1), 41–53 (2004)
7. Maymounkov, P., Maziéres, D.: Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. In: Druschel, P., Kaashoek, F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 53–65. Springer, Heidelberg (2002)
8. Rowstron, A., Druschel, P.: Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In: Liu, H. (ed.) Middleware 2001. LNCS, vol. 2218, pp. 329–350. Springer, Heidelberg (2001)
9. Rabin, M.O.: Fingerprinting by Random Polynomials. Technical Report TR-15-81, Center for Research in Computing Technology, Harvard University (1981)
10. Bloom, B.H.: Space/Time Trade-offs in Hash Coding with Allowable Errors. ACM Communications 13(7), 422–426 (1970)
11. Prakash, A., Starschenko, A., Chaparadza, R.: Auto-Discovery and Auto-Configuration of Routers in an Autonomic Network. In: SELFMAGICNETS 2010: Proc. of the International Workshop on Autonomic Networking and Self-Management, ICST ACCESSNETS 2010, Budapest, Hungary (November 2010)
12. Rétvári, G., Németh, F., Chaparadza, R., Szabó, R.: OSPF for Implementing Self-adaptive Routing in Autonomic Networks: A Case Study. In: Strassner, J.C., Ghamri-Doudane, Y.M. (eds.) MACE 2009. LNCS, vol. 5844, pp. 72–85. Springer, Heidelberg (2009)
13. Baumann, R., Heimlicher, S., Strasser, M., Weibel, A.: A Survey on Routing Metrics. TIK Report 262, Computer Engineering and Networks Laboratory, ETH-Zentrum, Switzerland (February 2007)
14. Stefanescu, H., Skrocki, M., Kuklinski, S.: AAODV Routing Protocol: The Impact of the Routing Metric on the Performance of Wireless Mesh Networks. In: Proc. of the 6th International Conference on Wireless and Mobile Communications, ICWMC 2010, Valencia, Spain (September 2010)
15. Kuklinski, S., Radziszewski, P., Wytrebowicz, J.: WARF: A Routing Framework for IPv6 based Wireless Mesh Networks. In: Proc. of the 2nd International Conference on Internet, ICONI 2010, Cebu, Philippines (December 2010)