Auto-Configuration of OSPFv3 Routing in Fixed IPv6 Networks

Alexej Starschenko
Fraunhofer FOKUS and TU Berlin
Berlin, Germany
starschenko@gmail.com

Nikolay Tcholtchev, Arun Prakash, Ina Schieferdecker
Fraunhofer FOKUS
Berlin, Germany
{*firstname.lastname*}@fokus.fraunhofer.de

Ranganai Chaparadza
IPv6 Forum, ETSI AFI
Germany
ran4chap@yahoo.com

***Abstract*—Today's Network Management highly depends on manual configuration of networking devices. On one hand, this leads to a high number of configuration related errors in the network, which must be compensated with additional OPEX (operational expenditure) effort. On the other hand, Network Management is not able to cope with the growing size of networks and the variety of devices due to manual configuration. This paper applies concepts from the area of Autonomic Network Management in order to develop components and algorithms for the auto-configuration of OSPFv3 routing in IPv6 based fixed network environments. A reference model for autonomic networks is applied for the design of the components. The developed key functionalities include: the automated computation and dissemination of OSPFv3 protocol configurations, topology discovery, automated partitioning of the network into OSPF areas, and in-band/in-network configuration and management of OSPFv3 routing. The proposed concepts are evaluated based on a prototype implementation and on measurements of its performance in the testbed of a large scale European project.**

***Index Terms*—Autonomics; IPv6; Auto-Configuration; Routing; GANA; OSPFv3; Autonomic Networking**

I. Introduction

The Internet plays a key role in today's life. In most areas of our society, services provided by it involve the global exchange of information. Even in critical areas, such as emergency services, medicine, etc. it plays a significant role. Thus the presence and proper functioning of the Internet is a crucial necessity, which has to be addressed by the telecommunications and computer science community.

The Internet is a global network, which consists of mostly *Internet Protocol (IP)* based networks that are operated by Internet Service Providers (ISPs). All these networks provide a set of basic networking functions, which other advanced networking services and functions rely on. When taking into consideration the impact of these services on today's life, the importance of Network Management as the domain that aims to provision and maintain smooth operation of these basic functions is clearly visible. This argument especially applies to the configuration of routing, a basic functionality that enables network communication over great distances (multiple hops) and allows eventually the global interconnection and usage of the Internet services.

The problems of current approaches in the domain of Network Management in general, and Configuration Management in particular, are well studied. On one hand, the aspect of manual configuration leads to a high number of errors in the network, and on the other hand, it is not able to cope with the growing size of networks and the variety of devices. The evolution of Network Management towards Autonomic Network Management and thus the evolution from Configuration Management towards Autonomic Configuration Management, defined by several initiatives [1] [2] [3] [4], promises to alleviate these problems by reducing the need for manual configuration. In that context, the main contribution of this work is to apply the concepts of Autonomic Configuration Management to the problems in the configuration of the OSPF routing protocol in IPv6 based fixed networks, and to identify and develop the components and algorithms required to achieve this.

The above considerations result in the following problem statement: Today, configuration of OSPF is done manually and this brings the problem of huge OPEX (operational expenditure) requirements due to the OSPF protocol complexity and the ever growing network sizes, especially when considering a multi-vendor environment. In addition to the problem of manual configuration, the dynamic adaptation (optimization) of OSPF such as OSPF-area partitioning is also done manually. Hence, it is required to come up with solutions that automate the configuration and the dynamic optimization of OSPF. IPv6 is a mature advancement of the IP protocol with inclusion of some features that could potentially enable the realization of Auto-Configuration of OSPF and dynamic optimization of OSPF routing areas. There is a need to find a way to combine these IPv6 features with autonomic network design principles in order to achieve Auto-Configuration and dynamic adaptation (optimization) of OSPF routing areas.

The rest of this paper is organized as follows: Section II presents the architectural aspects for OSPFv3 Auto-Configuration. The following section describes the process and the dynamic aspects of the Auto-Configuration framework. Section IV describes the implementation, based on which section V and section VI elaborate on the performed evaluations. The two sections that follow after discuss the obtained results and put the proposed framework in relation to similar research efforts thereby outlining similarities and differences. Finally, section IX draws conclusions and outlines future research directions.

978-1-4673-9283-9/15/$31.00 ©2015 IEEE

II. Architectural Components for Enabling Auto-Configuration of OSPFv3

To implement the proposed solution and provide the expected behaviour, a specification of the required components and the functionality expected from each of them is required. This section is intended to identify and justify the choice of the involved components and present the key functionality expected from each of them. Figure 1 shows a consolidated picture of identified components, on node and network level. The illustrated components are elaborated in the following subsections.

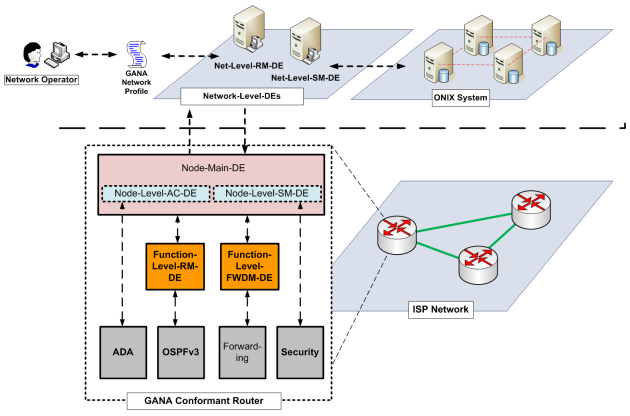

Fig. 1. GANA Conformant Network

Figure 1 presents the autonomic management architecture for autonomic management of OSPF and other managed networking functions that are closely linked to the problem addressed by the paper, namely forwarding, addressing and security. The architecture was derived on the basis of the GANA (Generic Autonomic Network Architecture) Reference Model [5] for autonomic networking and autonomic network management. The GANA Model is a hierarchical autonomic management framework that defines four basic levels of self-management (decision-making) at which autonomic-manager components, called Decision Elements (DEs) can be designed. DEs interact hierarchically in the sense that decisions that cannot be performed at a lower level should be taken at an intermediate upper level (i.e. as we go upwards, upper DEs control-loops become complex but slower than lower DEs control-loops). The GANA defines so called Managed Entities (MEs), e.g. network protocols, which are assigned to specific GANA Function-Level DEs, e.g. routing protocols of a node are assigned to the Function-Level-Routing-Management-DE (Function-Level-RM-DE in Figure 1) of a node. Thereby, the DEs are hierarchically ordered starting from 1) protocol in-built low-level DEs, which are to be seen as intelligent mechanisms within the network protocols and functional entities, continuing with 2) function-level DEs, which are autonomic agents managing important networking functions within a device, such as Routing, Forwarding, Mobility, 3) a node level DE (Node-Main-DE) that includes different sub-DEs responsible for managing various aspects of the overall node, and finally 4) network level DEs which execute autonomic behaviors on network wide scale. These different types of DE concepts provide the foundations for the design and specification of the OSPFv3 Auto-Configuration framework depicted on Figure 1.

A. Foundations of the Proposed OSPFv3 Auto-Configuration

The deployment of routing in a network is following a certain goal of the operator, which is determined by the use case of the network. This goal is expressed through network *Objectives*. Two such objectives are considered as important in the case of OSPFv3 Auto-Configuration: 1) OSPFv3 parameters and 2) Desired routing topology.These objectives should be defined in XML and encapsulated in a *GANA Network Profile*. A GANA Network Profile is provided by the network operator over the Network Governance interface on Figure 1. This GANA Network Profile is then merged with the GANA Capability descriptions of the devices in order to configure single routers and the network as a whole. A *GANA Capability description* can be seen as a self-description ”document” provided by the routers as they boot up. It describes the different features of a router including supported protocols, network addresses etc. The combination of the capability description and the network profiles results in concrete *GANA Node Configurations* for the routers, which are incrementally attached to the network.

Coming back to the above objectives: With the ”OSPFv3 parameters” objective, the operator can enforce that the OSPFv3 protocol-parameters will be set to a certain value network-wide. For example, he/she could decide to set timer and interval parameters, such as the Hello-Timer or the Router-Dead-Interval [6], to a low value in order to decrease the convergence time of OSPF, if high reliability is important for his/her network goals. Apart from the standard OSPFv3 protocol parameters, parameters which influence the autonomic OSPF-area partitioning should also be defined, e.g. size of the areas, number of areas, etc. It should be also considered that both objectives are optional and that the autonomic components in the network should be able to provide the full Auto-Configuration functionality with default values. The second objective defines the desired routing topology, which set once the corresponding physical topology, i.e. number of routers and intermediate links has been reached.The main role of the objectives is to keep the control over the Auto-Configuration process in the hands of the operator.

Looking at the devices capability descriptions in particular[7]: The *GANA Capability Description* published by the routers should contain vital information required for the computation (based also on the Network Profile) of an OSPFv3 configuration. This information can be considered rather static, as capabilities are not expected to change frequently during the runtime of the router. The items of information identified as belonging to the capabilities of a router are: 1) Addressing information (local and global IPv6 addresses), 2) Supported routing protocols (mainly whether OSPFv3 is supported), 3) Device- or protocol-vendor

information, 4) Hardware parameters, 5) Cost of running a certain protocol, e.g. a particular weight which reflects the operator's preferences regarding different features on a router, and 6) Neighbor information as provided by the IPv6 Neighbor Discovery (ND) protocol.

In addition, the Net-Level-RM-DE, which is the key component for the OSPFv3 Auto-Configuration, also publishes its capability description containing addressing information to facilitate its discovery by the routers.

Having drafted the artifacts (information models) required to perform OSPFv3 Auto-Configuration, the various components which process these artifacts need to be devised. The proposed solution as illustrated on Figure 1 requires the inter-working of distributed entities and players: operator, Net-Level-RM-DE and routers (DEs in the routers). The following sections aim to identify the mechanisms and protocols used by the proposed solution.

B. Information Exchange

The EFIPSANS project [3] has defined two extensions of IPv6, which can be used for the purpose of implementing the Auto-Configuration functionality. The basic idea of the first extension is to use the DHCP [8] servers, which are present in a high number in current networks, not only for address configuration and addressing related information, but also for the exchange of management information in general. For this purpose the ONIX (**O**verlay **N**etwork for **I**nformation e**X**change) [9] system was developed. The ONIX system - depicted on the upper right in Figure 1 - is designed to be a scalable, DHT (Distributed Hash Tables) based overlay network for the exchange of XML data. It supports advanced information exchange and manipulation functionality such as information publish, subscribe, notify, update and advanced search operations inside the data. Moreover as the ONIX system is designed as an extension of DHCPv6 servers, DHCPv6 has been extended to support these required operations [9]. Thus, ONIX is used for sending the *GANA Network Profile* from the operator to the Net-Level-RM-DE, to disseminate/share the *GANA Capability Descriptions* of the Net-Level-RM-DE and the routers, and to distribute the *GANA Node-Configuration Files* computed by the Net-Level-RM-DE.

Besides the ONIX-based information exchange the exchange of time-critical information between the DEs in the routers and the Net-Level-RM-DE is required, for instance for the notification of the Net-Level-RM-DE by the routers, after the computed node configuration was successfully applied. For this purpose an extension of the ICMPv6 protocol, specified in [10], is used. This extension defines a generic protocol for information exchange between entities (Decision Elements in this case) on different devices.

C. Network-Level-Routing-Management-DE

As described on Figure 1, the task of the Auto-Configuration of OSPFv3 is performed by a central autonomic manger component. In the context of GANA, which is the reference architecture used for the developed solution, the Network-Level-Decision-Elements are the components that play the role of the central controller in the whole network. For the Auto-Configuration of OSPFv3 routing the responsible Net-Level-DE is the **Net-Level-RM-DE** (see Figure 1). Based on the generic descriptions in the previous section, the following set of functions can be assigned to the Net-Level-RM-DE: 1) Self-Description and Self-Advertisement to ONIX, 2) Router Auto-Configuration, 3) Computation of the OSPF-area Partitioning 4) Auto-Discovery/Topology discovery based on the neighbor information derived from the IPv6 Neighbor Discovery cache and sent by each router, and 5) Network Governance Interface to the network operator.

D. Network-Level-Security-Management-DE

In addition, Figure 1 shows a Net-Level-SM-DE (Network Level Security Management DE) that is required in order to protect the process of OSPFv3 Auto-Configuration. The functionalities which are to be implemented by such a Decision Element were elaborated in [11], including aspects such as Intrusion Detection/Prevention as well as Authorization and Authentication for functional entities. In the scope of the current Auto-Configuration framework for OSPFv3, the Net-Level-SM-DE is mainly intended to authorize and authenticate network nodes and Decision Elements with respect to certain configuration actions, which should be executed only by trusted nodes/entities. In the scope of the EFIPSANS project [3] a Kerberos [12] based version of the Net-Level-SM-DE was implemented.

E. Node-Main-Decision-Element

Two key roles are defined by the GANA model for the Node-Main-DE on Figure 1. The first role includes functionalities that are essential for the bootstrapping of the node. To achieve this, the Node-Main-DE performs the **Initialization** and **Orchestration** of the underlying DEs during the boot-up of the router. In addition to this, the Node-Main-DE also provides the **Self-Description** and **Self-Advertisement** functionality for the router. This functionality consists of triggering the compiling of the *GANA Capability Description* of the router. This is a task of the Node-Main-DE because the Capability Description contains information about all DEs and protocols, and also information regarding various node attributes, which can be obtained only given the view on the whole node. This information gathering is based on a recursive process where the Node-Main-DE requests its MEs and underlying DEs for their capabilities. Thereby, each DE adds its capabilities and in turn delegates the request to its underlying DEs and MEs. Once all the capabilities have been aggregated, the Node-Main-DE obtains the neighbor information from the IPv6 ND protocol cache and adds it to the GANA Capability Description, in order to round it up with information concerning the point of the router's attachment to the network. Subsequently, the Node-Main-DE publishes the overall GANA Capability Description of the node to ONIX,

where it can be accessed by other network elements/entities subscribed for this data.

The other key role of the Node-Main-DE is to manage functionalities that are not mapped to any particular set of protocols and are required by every networking function. These networking functions that are identified by the GANA model are Auto-Configuration, Security-Management [11], Fault-Management [13] and Resilience-And-Survivability [5]. Inside a GANA conformant node these functions are introduced as four Node-Level-Sub-DEs, which are contained in the Node-Main-DE.

*Node-Level-AC-DE:*For the task of basic auto-configuration of OSPFv3, only the **Auto-Configuration** functionality of the Node-Main-DE can be considered as relevant. This functionality is performed by the *Node-Level-AC-DE* and is in charge of decomposing the *GANA Node Configuration*, which is given to the whole device, into Policies, Objectives and DE Configuration Data. Thereafter, the Node-Level-AC-DE distributes the relevant pieces of information to the corresponding entities. In the particular case of auto-configuring OSPFv3, the *GANA Node Configuration* contains only one single protocol configuration element, which is delegated to the Func-Level-RM-DE.

Node-Level-Security-Management-DE: A *Node-Level-Security-Management-DE (Node-Level-SM-DE)* - in the node zoom on Figure 1 - is needed as it is required to provide access control functionality during the auto-configuration phase for the devices. This functionality was discussed initially in [11] and is the node level counterpart to the authorization and authentication provided by the Net-Level-SM-DE (see section II-D). As previously mentioned, within the EFIPSANS project, a Kerberos based access control was evaluated.

F. Function-Level-Routing-Management-DE

The *Function-Level-Routing-Management-DE (Func-Level-RM-DE)* on Figure 1 is the manager component that must control and configure the OSPFv3 protocol on a router. For the Auto-Configuration functionality the main task of this DE is to apply the protocol configuration computed by the Net-Level-RM-DE and sent to the DE via the decomposition process of the Node-Level-AC-DE. When the configuration is received, the Func-Level-RM-DE needs to impose that configuration and start the OSPFv3 routing functionality on the device. In case OSPFv3 is already running, the DE has to reconfigure it according to the newly received configuration. Moreover, the Net-Level-RM-DE also needs to be notified by the router when the configuration is successfully applied in order to avoid OSPFv3 inconsistencies and disruptions. Thus, after the (re-)configuration process of the OSPFv3 protocol has been initiated, the Func-Level-RM-DE needs to monitor the configuration state of the protocol and inform the Net-Level-RM-DE about the outcome. Finally, the Func-Level-RM-DE needs to contribute its capabilities to the Capability-Description of the overall device (router). The most important information that has to be provided by the Func-Level-RM-DE relates to routing protocols and their protocol-version, supported by the device, and the vendor information of the device as a whole as well as for individual protocols. This is required for example in case of software-routers, where routing protocols can be provided by different routing software vendors. This information is needed by the Net-Level-RM-DE to obtain the correct vendor specific configuration for OSPFv3 for a particular device.

G. Additional Components

Besides the above described components, Figure 1 shows two additional modules which complete the picture of a router Auto-Configuration. These are *Autonomic DHCP Architecture (ADA)* [14] and the function level Forwarding Management DE (Function-Level-FWDM-DE). In general, ADA would be responsible for the address configuration of a router, a task which would precede the OSPFv3 Auto-Configuration process described here. Some more details on ADA are provided in the section elaborating on related research efforts. Furthermore, the Function-Level-FWDM-DE would be responsible for (re-)configuring the Forwarding plane of a router, e.g. Access Control Lists. This type of configuration would normally require an established routing and would follow the OSPFv3 Auto-Configuration. These two components were not integrated within the current prototype, however they constitute useful complements to the concepts presented here.

III. The Process of OSPFv3 Auto-Configuration

The main goal of the following process description is to provide a detailed understanding of the Auto-Configuration steps and the overall technical behavior of the developed components. In order to best elaborate, the scenario is split into a set of steps:

Step 1: Initial Deployment: The first step of the scenario is the deployment of the key components illustrated on Figure 2. The main component is the Net-Level-RM-DE, as it performs and controls the overall Auto-Configuration process in the network. Moreover, an interaction between the operator and the Net-Level-RM-DE is required to allow the operator to impose the network objectives on the Net-Level-RM-DE before the deployment of the network can start. Thus, the operator terminal/console should be also initially placed in the network. The ONIX system (refer Section II-B) is used to provide the functionality for the exchange of most data during the Auto-Configuration of OSPFv3, and should also be deployed during the first step of the scenario. As no routing is present at this stage, the initially deployed components have to be directly connected, as shown in Figure 2.

Step 2: Network Governance: The second step of the scenario also belongs to the initial phase, before the deployment of the routers, and can be considered as the *Network Governance Phase*. The main idea of this is to shift the configuration effort of the operator from traditional device centric perspective to the new network-centric configuration perspective. Thus in this step, the operator delegates high-level routing related objectives to the Net-Level-RM-DE, since during the deployment of the routers this DE will configure

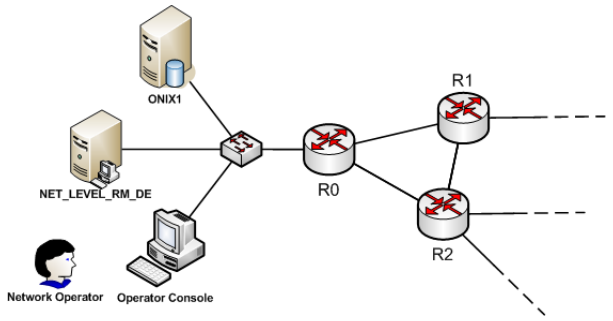

Fig. 2. Initial Deployment

the routers without the need for any human involvement. For the Auto-Configuration of OSPFv3, two objectives are used: 1) OSPFv3 parameters and 2) Desired routing topology. The first one defines default values for OSPF parameters for the whole network. In addition to this, the first objective also specifies desired partitioning parameters, such as the threshold number of routers for an unpartitioned network/area. Via the second objective, the operator defines the desired OSPFv3 topology for the network, which he/she plans to deploy. The OSPF partitioning of the desired topology is the main significant configuration information and will be applied by the autonomic components in the network when the desired topology is reached, i.e. when the planned number of routers is deployed and connected in the corresponding physical topology.

Step 3: Deployment of Routers Before Partitioning: After the initial phase, the operator starts building the network by adding routers until the desired topology is reached. The first router is attached to the initially deployed components, so that the Net-Level-RM-DE, the other initially deployed components and the first router are directly inter-connected and can communicate without routing. After its placement in the network, the router and its autonomic components (DEs) are automatically started. Iteratively, further routers are placed, interconnected with the already present network and bootstrapped. During the bootstrapping of a router, the (auto-)configuration of its interfaces is required to enable the communication processes that are taking place during Auto-Configuration of the OSPFv3 routing protocol. After the bootstrap, the Net-Level-RM-DE first discovers the point of attachment of the new router. Further, the Net-Level-RM-DE needs to determine the capabilities of the router, including the vendor information of the router, since the Net-Level-RM-DE should be able to configure devices from different vendors. Based on the discovered information, the DE computes the configuration for the OSPFv3 protocol of the new router (*GANA Nodeconf*, refer Section II) and sends it to the router where it is applied to the protocol/entity, by the Func-Level-RM-DE. In order to include the new router into the OSPFv3 network, the configuration and activation of the OSPFv3 protocol on the newly added device is not sufficient. The OSPFv3 configuration of the routers which are connected to it also has to be updated by the Net-Level-RM-DE. For the configuration of OSPFv3 on each router, the parameters according to objectives passed to the Net-Level-RM-DE are used. Because of the initially small size of the network during this step of the scenario, all routers are at first automatically configured as area-0 routers until the number of routers in this area crosses the threshold defined by the operator in the routing objectives. This process is repeated for each deployed router.

A key feature of the proposed OSPFv3 Auto-Configuration process is that all management communication in the scenario is *in-band* communication. This means that no dedicated management network is required and the components have to use the network, which is being configured, for the Auto-Configuration related communication.

Step 4: OSPF Network Partitioning: Partitioning of the network into OSPF areas is an essential requirement for the proper functioning of the OSPFv3 protocol. The Net-Level-RM-DE provides this functionality during the Auto-Configuration process of the network. This means that when the network reaches a certain size (number of routers) while it is deployed by the operator, the DE has to divide the OSPFv3 topology into OSPF areas, compute new configurations for the routers according to the partitioning and apply it to the network. The recomputed router configuration, which is based on the outcome of the partitioning algorithms - e.g. k-neighborhood as provided by the JUNG (Java Universal Network/Graph Framework) [15] library, is distributed and applied to all routers in the network. The threshold that expresses the maximum size of an unpartitioned network (or area) is considered to be given by the operator, since it depends on the goals of the network. The usage of a default value derived from OSPF properties is also considered.

Step 5: Deploying Routers After Partitioning: At this state of the process, an OSPFv3 configuration which contains multiple OSPF areas is present in the already deployed network and the operator is adding and bootstrapping new routers to reach his/her desired topology. The behaviour of the Net-Level-RM-DE depends on the size of the area the new router is attached to. If the size including the new router does not exceed the unpartitioned network/area threshold, OSPFv3 is auto-configured on the new router and on the affected routers, such that it includes the router in that area. If the threshold is exceeded, the OSPFv3 topology of the whole network is re-partitioned by the Net-Level-RM-DE and the new configuration is applied to all routers.

Step 6: Desired Topology is Reached: The last step of the scenario begins when/if the deployed network reaches the routing topology planned by the operator. In this case, the Net-Level-RM-DE applies the desired routing configuration, which the operator has specified via the routing objectives delegated to the DE during the initial phase of the scenario, to the deployed network. For this, the DE reconfigures OSFPv3 in the network according to the OSPFv3 partitioning objectives set by the network operator in the desired topology objective.

Tool/Language	Required for the Functionality/Element
JAVA (1.6)	Main programming language; implementation of all DEs, OSPFv3 partitioning algorithm, and Auto-Configuration framework.
Chaco	Software for Partitioning Graphs. It is used by the autonomic OSPFv3 partitioning algorithm.
XORP (1.5) [16]	A software routing platform for Linux, which provides OSPFv3. It is the target of the auto-configuration processes.
Quagga (0.99.12) [17]	Another software routing platform. It was the first choice for the developed solution, but was dropped after some problems were identified, as described in the next section.
Radvd [18]	A IPv6 Routing Advertisement daemon for Linux. It is started by the Func-Level-RM-DE and enables the Auto-Configuration of the default gateway on newly attached routers.
C, JNI	JAVA Native Interface and C were used to port Chacos API to JAVA.
JUNG	A JAVA library which is used to store, manipulate and partition graphs. It is used for the topology and topology graph handling/partitioning. [15]
JAVA RMI	The JAVA Remote Method invocation is used for the in-node communication between the DEs. For this, each DE exports the defined interfaces to an RMI server, which then can be accessed by the other DEs in the node.
Perl	Scripting language that was used to implement operating system related methods, which are not directly accessible from JAVA.

TABLE I
TOOLS AND PROGRAMMING LANGUAGES USED FOR THE PROTOTYPE

IV. IMPLEMENTATION OF THE OSPFV3 AUTO-CONFIGURATION FRAMEWORK

This section is intended to introduce the experiences related to the implementation of the proposed framework. The evaluation of the solution in Section V is based on the prototype presented here.

Tools and Technical Details: The scope of the implementation was to develop components to provide the Auto-Configuration of OSPFv3 on Linux soft-routers. Table I provides a list of the most important tools and programming languages, which were used to fulfill this goal. Besides these tools and languages, components were used which have been developed in the context of the EFIPSANS Project [3], such as the ONIX system and the ICMPv6 extensions.

Main Issues Faced and Lessons Learned: In the following, problematic issues are presented which were faced during the implementation of the prototype. It not only explains some development decisions, as they were made to alleviate these problems, but also provides a picture of the implemented prototype.

The first problem is related to the Auto-Configuration of OSPFv3. When auto-configuring multiple ($\geq$ 3) instances of OSPFv3 simultaneously, LSA oscillations were occurring, which means that the routers were permanently exchanging LSAs without converging. A solution is provided by the realization of a strict iterative process that ensures that OSPFv3 is not auto-configured on more than two routers at the same time.

The second major problem was experienced with the Quagga software routing platform [17], chosen initially as the routing platform to experiment the Auto-Configuration process. Quagga (0.99.12) does not fully support OSPFv3 areas and some routes were missing after the Quagga routers were auto-configured according to the computed OSPFv3 area partitioning. In the case where the routes to the Net-Level-RM-DE or ONIX were missing, the auto-configuration of OSPFv3 could not proceed further. The only solution to this problem was the migration to XORP [16] as another routing platform. Given the vendor support implemented as part of the Auto-Configuration framework, this migration could be simply done by the provisioning of the according configuration data in the framework and minor extensions of the Func-Level-RM-DE to manage XORP.

Another problem was experienced with both routing platforms during the re-configuration or deactivation of OSPFv3 by using the CLI. It could be observed that the dynamic change of some of the OSPFv3 parameters on the running platform led to unexpected errors and exceptions. In general this also leads to the conclusion that to be able to deploy autonomic solutions for current protocols and resources, further tests of the existing protocols and solutions is most certainly required. To solve these problems in the implemented solution, the whole routing platform is restarted for the re-configuration of OSPFv3 and stopped for its deactivation.

V. EVALUATION: TESTBED CASE STUDY

The proof of the concepts introduced in the proposed solution, which has been developed in this work, is done by the implementation of a prototype (refer Section IV) and its evaluation. First the environment used for the case study is introduced. Thereafter, the evaluation of the overall process of Auto-Configuration of OSPFv3 and of the autonomic OSPFv3 partitioning are presented.

A. Testbed

For their evaluation, the developed components were deployed on a testbed that was built in the course of the large scale EFIPSANS EU project. This testbed is a network consisting of virtual machines and normal PCs. It includes **11 routers** and further nodes, on which the other required components are placed. Figure 3 shows the testbed topology including important hardware parameters.

Linux was installed as the operating system on all network elements in the testbed and the developed Auto-Configuration components deployed on them. As defined in the process description in section III and shown in Figure 2, the initially required components, namely the Net-Level-RM-DE, ONIX and the operator console, are deployed on-link with each other. As it can be seen in the testbed topology in Figure 3, the Net-Level-DE and the operator console are placed on

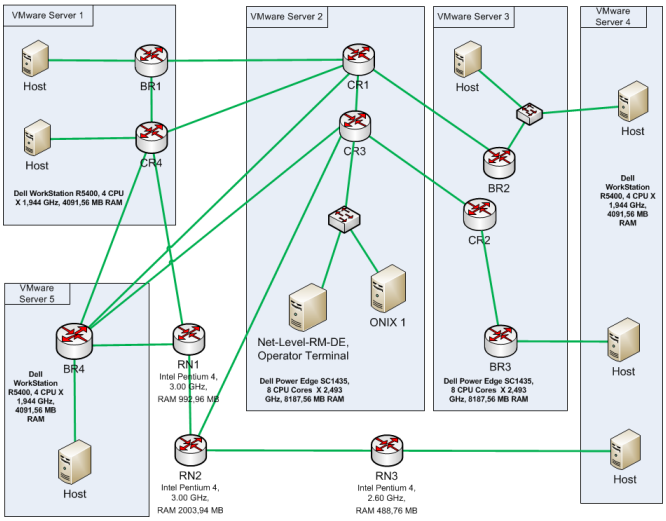

Fig. 3. Testbed Network

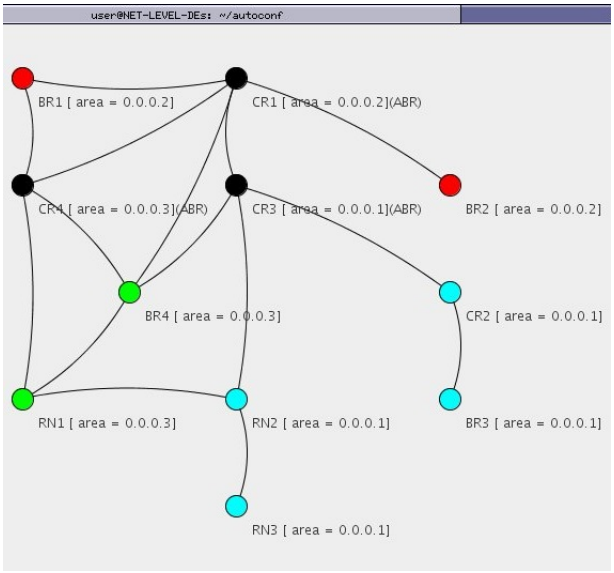

Fig. 4. OSPFv3 Configuration in the Testbed

a single virtual machine on the second VMware server and connected to a virtual switch. Also connected to this switch, on a separate virtual machine, there is an ONIX server deployed. The remaining virtual machines and the three PCs play the role of routers or hosts. The host machines are used in the scenario only to demonstrate the connectivity and routing in the testbed network. On each of the machines performing the role of a router, the node components developed for the Auto-Configuration of OSPFv3 are placed, namely the Node-Main-DE, Node-Level-AC-DE, Func-Level-RM-DE and Func-Level-FWDM-DE. Additionally, the XORP routing software is installed. The router CR3 is the one on-link with the initial components (compare to R0 in the scenario Figure 2) and must be bootstrapped first to get auto-configured.

B. Case Study

To exemplify the partitioning and the final outcome of the Auto-Configuration process on the testbed topology, Figure 4 shows a Chaco [19] screen shot of the view of the Net-Level-RM-DE on the configured and partitioned OSPFv3 topology. This is the result of the Auto-Configuration performed on the whole testbed. The starting order of the components on the routers and thus the configuration order performed by the Net-Level-RM-DE was: *[CR3,CR1,BR1,CR4,BR4,BR2,CR2,BR3,RN1,RN2,RN3]* (see Figure 3). It can be observed that the Net-Level-RM-DE has partitioned the OSPFv3 network into three areas and inter-connected them with an area-0. This configuration was further verified by checking the OSPFv3 LSDB (Link State Database) of the XORP platform on the routers, as done for all the other configurations during the measurements.

VI. Evaluation: Measurement Results and Complexity Assessment

The next evaluation of the prototype consists of the performance measurements of the Auto-Configuration of OSPFv3 on the routers in the presented testbed. The first measured metric is the **time required for the Auto-Configuration of OSFPv3 without partitioning**. This metric was assessed by performing the auto-discovery and Auto-Configuration process by iteratively adding more routers to the network, as described in the process in section III. The time between the initial start of the components and the point in time where OSPFv3 was configured and converged in the network was measured, i.e. notification of the successful configuration were received from all routers by the Net-Level-RM-DE. The measurement was recorded from the start of the Net-Level-RM-DE as the main component, which then initially requests and processes the network profile and its related data. Then, one after the other the Node-Main-DEs on the bootstrapped routers were started. The starting order of the components on the routers and thus the configuration order performed was the same as the one in the scenario section above. As this measurement is intended to show the performance of the components without the partitioning, the area size threshold was set to a high value and all OSPFv3 routers are configured as part of area-0. The results of the measurement can be seen in the graph in Figure 5. The most important observation is that the Auto-Configuration time is **linearly dependent on the size of the network**. In the testing environment, the Auto-Configuration of OSPFv3 on all eleven routers was performed in 440 seconds and the average-time for the Auto-Configuration of a newly attached router from the moment of its activation was 40 seconds. These performance measurements have been conducted on the specific testbed topology. However, the results are still considered significant, as most of the components and algorithms are topology agnostic.

It is difficult to compare the performance of this approach to traditional network management techniques, such as SNMP and CLI, because of the human involvement in these traditional

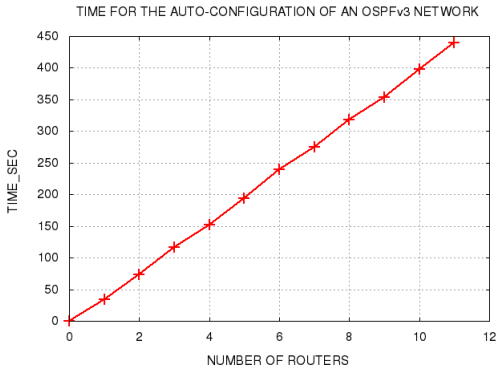

Fig. 5. Measurement: Auto-Configuration of OSPFv3

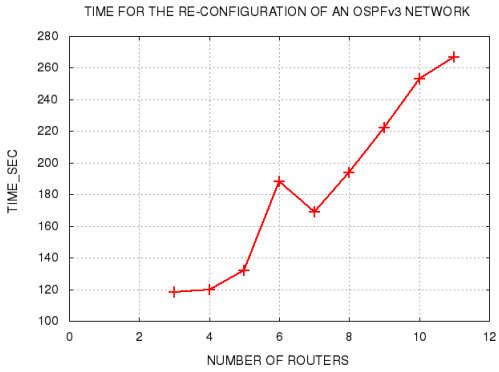

Fig. 6. Re-Configuration of an OSPFv3 Network

methods. Still it can be assumed that a configuration time of 40 seconds cannot be achieved in general by manually issuing configuration commands to the device or even by scripting.

The second performance metric which can be assessed in the testbed is the **re-configuration time of an OSPFv3 network** required to apply a computed partitioning. In general, it can be assumed that the performance of the re-configuration has the same complexity as the initial configuration, as similar steps are done by the algorithms. For the measurement of the re-configuration time of networks from 3 up to 11 routers, the area size threshold was set to the appropriate size and the Auto-Configuration process was performed. After OSPFv3 was configured, and the components on the last router were started, the crossing of the threshold was identified by the Net-Level-RM-DE and partitioning and re-configuration was performed. In this measurement, the time for computing the new partitioning was not included; thus the measured time spans from the beginning of the deactivation process of OSPFv3 in the network until the partitioned configuration was applied and confirmed by all routers. The outcome of the measurement can be seen in Figure 6. As expected, an almost linear performance can be observed also for the re-configuration process. Still, a higher spread of the values can be seen, compared to the initial configuration measurements (see Figure 5). This effect can be attributed to the impact of the partitioning outcome on the configuration order and the number of affected routers on the re-configuration operations. However, this impact can be limited by the maximum degree of a node in the topology graph (as our experiences indicate) and thus has no implication on the linear complexity. The second observation that can be made given these measurement results is that in general, the re-configuration process, with the average time of 24 sec for a router, is faster than the initial OSPFv3 configuration. The main reason for this difference is the time that is required for the bootstrapping of the routing software and the other initial operations performed by the DEs, such as the computation of the capability description during the first configuration of the router. This bootstrap time even exceeds the OSFPv3 deactivation time during the re-configuration of a router.

The two measurements of the Auto-Configuration performance of the implemented prototype lead to the following assessment: both the initial configuration and the re-configuration are in the complexity class $O(N)$ (N = number of routers). In the worst case, partitioning could be triggered after the attachment of every new router. This is the case if the outcome of every partitioning contains an area of the size of the *area-size* threshold and the newly added router is always attached to this area. In this case, the performance of the overall Auto-Configuration of OSPFv3 process can be estimated with the complexity $O(N^2)$.

VII. Discussion and Analysis

The main goal of the developed components is the reduction of the Operational Expense (OPEX) incurred by the operators for the configuration of OSPFv3 routing in their networks. The key observation which can be pointed out when analyzing the OPEX implication of the presented approach is the shift from device-centric to a network-centric configuration. During the runtime of the Auto-Configuration process, the only involvement of the operator is the physical deployment and starting of a router. After this step the auto-discovery and Auto-Configuration performs all required configuration tasks. Thus, a significant OPEX reduction can be observed compared to the traditional network management approaches, in which all configuration tasks have to be (at least partially) performed by the operator manually.

One of the approaches to numerically assess OPEX is to consider the number of commands which have to be issued during the configuration process. In the scenario presented here (section V), the number of CLI commands which is required to configure a new XORP router (with 4 interfaces) is 51. To configure the network of n routers with i interfaces, in average $n \times (15 + i \times 9)$ commands were needed. During the runtime of the Auto-Configuration, the number of commands is obviously zero, as no intervention of the operator is needed.

Although no human involvement is required during the router configuration process, the effort for the configuration of the Auto-Configuration components themselves must be taken into account for the OPEX assessment. As described in the

proposed solution, the configuration of the autonomic components which perform the Auto-Configuration (Net-Level-RM-DE) is done over the Network Governance Interface which is implemented by the GANA Network Profile. Thus, the profile design and creation has to be done by the operator, instead of the configuration of individual devices. The profile design, when rusticated to the configuration of OSPFv3, is composed of the following tasks: 1) Creation of the GANA Network Profile structure, 2) Specification of OSPFv3 parameter values and OSPFv3 partitioning parameter values (optional), and 3) Desired OSPFv3 routing topology definition (optional). This process has to be performed once before starting the actual router deployment. When taking into account only the profile creation and the parameters population and leaving the topology design to the autonomic components, the OPEX effort of the profile-design is constant ($O(1)$), whereas the OPEX of the traditional approach is linearly increasing with every new router ($O(n)$; with n=number of routers). Given this analysis, the OPEX reduction introduced by the autonomic solution is significant ($O(1)$ vs. $O(n)$). The effort for the design of the desired topology, which can be given to the Net-Level-RM-DE by the operator to fully control the outcome of the auto-configuration process, when seen from the OPEX perspective, is similar to the traditional approach ($O(n)$), as the operator can influence the configuration of individual devices. Still, from the auto-configuration perspective this part is not required, and was defined mainly to maintain the possibility for the operator to have full control over the configuration of the individual devices.

VIII. Related Work

A comprehensive overview of traditional Network Management and especially Configuration Management practices, protocols and tools is given in [20].The *Simple Network Management Protocol (SNMP)* [21] makes use of MIBs (Management Information Base) to implement the management functionality. It provides methods to remotely read and write the MIBs on the managed resources, and thus to monitor their state and to control their configuration. It is mostly used for monitoring tasks, as it provides advanced MIB state notification mechanisms, but also for other Configuration Management operations. The problem with SNMP is rooted in its dependency on standardized MIBs and the fact that some of the parameters that need to be configured and controlled are vendor specific and not part of any standardized MIB. This means that these parameters cannot be managed through SNMP and their management relies on other tools, such as the *CLI (Command Line Interface)* of the device to be configured. The CLI defines a console based interface to the device, through which the state of the resource can be manually requested and (re-)configured. Finally, a modern protocol for Network Management is constituted by the *Network Configuration Protocol (NETCONF)* [22] that is built upon technologies from the WWW domain, such as XML. NETCONF is mainly targeting the configuration related tasks of Network Management. It allows a hierarchical structuring of configuration information through XML based NETCONF datastores, and provides operations to remotely modify this configuration data. Typically the contents of the NETCONF structure are defined using CLI, thus in this case it acts as a protocol to structure and remotely execute CLI configuration requests. But as in the case of SNMP and CLI, NETCONF also relies on the involvement of the operator, as it does not provide mechanisms for the *auto-discovery* of incrementally connected routers, as implemented in the current work.

Next, the some related research efforts from the past years are elaborated. A comparable approach is introduced by the *Autonomic Network Management Architecture (ANEMA)* [2]. ANEMA mainly focuses on Network Management through the definition and deployment of network objectives/goals as utility functions. This aspect can be compared to the policies and objective that are passed to the DEs using the Auto-Configuration framework developed as part of the proposed solution (refer Section II). The advantage of the approach chosen in this paper is that it is based on the GANA Reference Model, which covers additional requirements for autonomic network management, such as standardizable [4] interfaces and communication facilities that are not clearly defined in ANEMA. Moreover, the Auto-Configuration framework proposed in this paper does not constrain the definition of objectives and policies, and thus could encapsulate the objectives as realized by ANEMA.

Another interesting functionality that is discussed in the scientific community and has resulted in some IETF standards is the Auto-Configuration of IPv6 addresses. When focusing on the aspect of the Auto-Configuration functionality, this work is closely related to the solution elaborated here. The most widely known Auto-Configuration practice is defined by the *Stateless Address Auto-Configuration* of the IPv6 protocol [23]. However, the methods defined by this process differ from the ones elaborated for the Auto-Configuration of OSPFv3. When taking a look at the Auto-Configuration functionality, the main difference is that the *Stateless Address Auto-Configuration* is performed locally between router and a neighbouring host, whereas for the Auto-Configuration of OSPFv3, a central component with a complete network view is required. In addition, IPv6 comes with the concept of *Statefull Auto-Configuration*, which relies on a centralized on-link DHCPv6 server to provide information to hosts. In that context, the current approach extends this practice to the Auto-Configuration of OSPF routers and the belonging areas required for effective routing. To auto-configure IPv6 addresses of routers, an interesting approach using a DHCPv6 server on a distance of multiple hops is defined by the *Autonomic DHCP Architecture (ADA)* [14].

With respect to the partitioning of an OSPFv3 network into areas, the authors of [24], [25] define algorithms for partitioning of OSFP in wireless networks. The partitioning algorithms elaborated in this work are based on this work. However, the algorithms have been adopted for fixed network environments and have been extended with respect to the identified requirements.

Another important functionality of the proposed solution is the discovery of the IP layer network topology. Several approaches can be found to provide this functionality, e.g. as specified in [26], [27]. The main difference is that these methods perform the topology discovery using the SNMP protocol, whereas the solution defined in this paper uses the inbuilt features of IPv6 to retrieve the required topology information. Thus, the proposed solution tries to utilize as far as possible the inbuilt features of IPv6 instead relying on other or introducing new protocols and technologies.

IX. Conclusions and Future Work

A new approach for the configuration of the OSPFv3 routing protocol has been presented in this work. This approach alleviates the well known problems of the current approaches, which are mainly rooted in the need for human involvement in the configuration of every individual router. The development of the solution proposed in this work is based on the GANA reference model and IPv6 including some recently proposed extensions. Components and processes were identified and developed, which provide the following functionality: 1) They allow the network operator to define and impose high level OSPFv3 routing objectives to the network, 2) An autonomic manager component (Network-Level-Routing-Management-DE) performs the Auto-Discovery and Auto-Configuration of routers incrementally attached to the network, 3) Based on the Capability Description published by each router, their point-of-attachment in the network, and the objectives defined by the operator, the Net-Level-RM-DE computes the appropriate configuration of the OSPFv3 protocol and applies it to the router, and finally 4) As part of this process automated OSPFv3 partitioning is performed by the Net-Level-RM-DE, which ensures that the number of routers in an OSPF area in the network do not cross a certain threshold. The evaluation of the proposed concepts has shown that through the application of the proposed solution a significant reduction in OPEX can be gained during the configuration of OSPF in particular, and in Network Management in general.

Although a complete solution has been presented, developed and evaluated in this work, some potential improvements and extensions have been identified and discussed during the presentation of the implementation. This includes more promising solutions for address auto-configuration of routers and for ONIX discovery, as well as the need for a more elaborated mapping mechanism for vendor specific parameters within the Auto-Configuration framework.

References

[1] "An architectural blueprint for autonomic computing." IBM, White Paper Fourth Edition, 2006.

[2] H. Derbel, N. Agoulmine, and M. Salaün, "ANEMA: Autonomic network management architecture to support self-configuration and self-optimization in IP networks," *Comput. Netw.*, vol. 53, no. 3, pp. 418–430, 2009.

[3] "EC FP7-IP EFIPSANS Project," www.efipsans.org, 2008-2010, INFSO-ICT-215549.

[4] "ETSIAFI Autonomic Future Internet ISG," portal.etsi.org/portal/server.pt/community/AFI/344, 2013.

[5] R. Chaparadza, "Requirements for a Generic Autonomic Network Architecture (GANA), suitable for Standardizable Autonomic Behavior Specifications for Diverse Networking Environments," *International Engineering Consortium (IEC), Annual Review of Communications*, vol. 61, 2008.

[6] J. Moy, "RFC 2328: OSPF Version 2," IETF, Tech. Rep., 1998. [Online]. Available: www.ietf.org/rfc/rfc2328.txt

[7] A. Prakash, A. Starschenko, and R. Chaparadza, "Auto-Discovery and Auto-Configuration of Routers in an Autonomic Network," in *SELF-MAGICNETS 2010 in conjunction with AccessNets 2010*, 2011.

[8] M. Carney and C. Perkins, "Dynamic host configuration protocol for ipv6 (dhcpv6)", rfc 3315," 2003.

[9] R. Chaparadza, R. Petre, A. Prakash, F. Németh, S. Kukliński, and A. Starschenko, "IPv6 Features and Extended IPv6 (IPv6++) that enable Autonomic Network Setup and Operation," in *SELFMAGICNETS '10: Proceedings of the International Workshop on Autonomic Networking and Self-Management in the Access Networks*, 2010.

[10] R. Chaparadza, R. Petre, S. Cheng, L. Xin, and Y. Li, "ICMPv6 based Generic Control Protocol (IGCP)," Internet Engineering Task Force, Internet Draft, Mar. 2010. [Online]. Available: http://tools.ietf.org/html/draft-chaparadza-6man-igcp-00

[11] Y. Rebahi, N. Tcholtchev, R. Chaparadza, and V. Merekoulias, "Addressing security issues in the autonomic future internet," in *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*, 2011, pp. 517–518.

[12] C. Neuman, T. Yu, S. Hartman, and K. Raeburn, "The Kerberos Network Authentication Service (V5)," RFC 4120 (Proposed Standard), Internet Engineering Task Force, July 2005, updated by RFCs 4537, 5021. [Online]. Available: http://www.ietf.org/rfc/rfc4120.txt

[13] N. Tcholtchev and R. Chaparadza, "Autonomic fault-management and resilience from the perspective of the network operation personnel," in *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, 2010, pp. 469–474.

[14] C. Simon, F. Nmeth, F. Uzsk, G. Rtvri, F. Ficsor, and R. Vida, "Autonomic DHCPv6 Architecture," in *The 3rd IEEE International Workshop on Management of Emerging Networks and Services (GC'11 Workshop - MENS'11)*, Houston, Texas, USA, Dec. 2011.

[15] "JUNG - Java Universal Network/Graph Framework," http://jung.sourceforge.net/.

[16] "eXtensible Open Router Platform," http://www.xorp.org/.

[17] "Quagga Routing Suite," http://www.quagga.net/.

[18] "Linux IPv6 Router Advertisement Daemon (radvd)," http://www.litech.org/radvd/.

[19] B. Hendrickson and R. Lelandy, "The Chaco User's Guide Version 2.0," Sandia National Laboratories, Albuquerque, NM 87185-1110, Tech Report SAND95-2344, July 1995.

[20] A. Clemm, *Network Management Fundamentals*. Cisco Press, 2006.

[21] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "Simple Network Management Protocol (SNMP)," RFC 1157 (Historic), Internet Engineering Task Force, May 1990. [Online]. Available: http://www.ietf.org/rfc/rfc1157.txt

[22] R. Enns, "Netconf configuration protocol," Internet Engineering Task Force, RFC 4741, Dec. 2006. [Online]. Available: http://tools.ietf.org/html/rfc4741

[23] S. Thomson and T. Narten, "IPv6 Stateless Address Autoconfiguration," RFC 2462 (Draft Standard), Internet Engineering Task Force, December 1998, obsoleted by RFC 4862. [Online]. Available: http://www.ietf.org/rfc/rfc2462.txt

[24] S. Galli, H. Luss, J. Sucec, A. McAuley, S. Samtani, D. Dubois, K. DeTerra, R. Stewart, and B. Kelley, "A Novel Approach to OSPF-area Design for Large Wireless ad-hoc Networks," in *2005 IEEE International Conference on Communications, 2005. ICC 2005*, vol. 5, may. 2005, pp. 2986 – 2992.

[25] J. Sucec, J. Unger, K. Chang, S. Samtani, B. Russell, B. Biagini, and A. Staikos, "Evaluation of an automated ospf area design utilty for wireless battlefield networks," in *Proceedings of the 2006 IEEE conference on Military communications*, ser. MILCOM'06. Piscataway, NJ, USA: IEEE Press, 2006, pp. 2895–2901. [Online]. Available: http://dl.acm.org/citation.cfm?id=1896579.1897019

[26] Y. Breitbart, M. Garofalakis, B. Jai, C. Martin, R. Rastogi, and A. Silberschatz, "Topology discovery in heterogeneous IP networks: the NetInventory system," *IEEE/ACM Trans. Netw.*, vol. 12, pp. 401–414, June 2004.

[27] H.-C. Lin, S.-C. Lai, and P.-W. Chen, "An algorithm for automatic topology discovery of IP networks," *ICC 98 1998 IEEE International Conference on Communications Conference Record Affiliated with SUPERCOMM98 Cat No98CH36220*, pp. 1192–1196, 1998.