View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

Paper

On IPv6 Experimentation in Wireless Mobile Ad Hoc Networks

Monika Grajzer[1,2] and Mariusz Głąbowski[2]

[1] *Gido Labs Sp. z o.o., Poznan, Poland*
[2] *Chair of Communications and Computer Networks, Poznan University of Technology, Poznan, Poland*

Abstract—With the growing interest towards the Internet of Things IPv6-based mobile ad hoc networks (MANETs) become a key enabling technology offering the possibility of automated, unsupervised network configuration and operation. Such a functionality calls for an accurate and reliable testing of the newly proposed solutions, which is challenging due to the dynamic, decentralized and ad hoc nature of MANETs. In this work selected topics are presented on performing IPv6 protocols experimentation in wireless, IPv6-only mobile ad hoc networks – including both simulation – and testbed-based evaluation. Based on the authors experience with the evaluation of the extended IPv6 Neighbor Discovery protocol (ND++) proposed during the course of research, the selection of an open-source simulation environment is presented and a comparison between simulation and emulation experimentation methods is provided. Benefits and drawbacks of both these methodologies for testing IPv6 solutions are depicted. Moreover, the important aspects of topology and mobility considerations are considered. Finally the authors propose a testing approach that would allow for a detailed and accurate evaluation by means of open-source, easily accessible and low-cost methodologies.

Keywords*—*IPv6 simulation, IPv6 wireless testbed, MANET experimentation, MANET protocols evaluation, Neighbor Discovery ++ (ND++), NS-3.

1. Introduction

The features of mobile ad hoc networks (MANETs), which allow them to adapt, dynamically follow the changing networking environment and perform well without a pre-established infrastructure, make them an ideal basis for the Internet of Things. Accompanied with the IPv6 protocol stack, IPv6-based MANETs constitute a perfect solution for bringing Future Internet into the world of connected devices [1], [2]. Internet of Things, though, calls for an automated, unsupervised network configuration and operation [2], [3], since it is expected to ensure sustainable network functionality with minimal external supervision. This implies the need for a detailed, accurate and reliable testing of the newly proposed solutions. Especially stateless address autoconfiguration, as a means of a "plug and play" network set-up, is among the key IPv6 mechanisms that require thorough testing in many realistic and demanding MANET scenarios. This allows to ensure their performance at the level corresponding to the high users and network maintenance expectations.

Testing the newly proposed IPv6 solutions usually is a two-step approach: at the first stage the research idea is evaluated in the course of simulations, secondly the real-world evaluation at the testbed platforms and field trials are performed. Simulations of MANET networks can be executed by means of several available network simulators, including commercial (e.g. OPNET/Riverbed [4] or QualNet [5]) and open-source ones (e.g. OMNET++ [6] or NS-3 [7]), as well as the in-house simulators created to address particular needs of the conducted research. The second step, requiring close to real world conditions, is very often having a pre-commercial character. In case of testing IPv6 networks it can be performed on the big testbed platforms, including those certified with IPv6Ready logo [8], [9]. These are, however, usually designed as the fixed networks environment. Due to the very specific nature of mobile ad hoc networks these testbeds are not suitable for testing most of the MANET-dedicated IPv6 solutions, which leaves MANET researchers and developers at the difficult position. There are few available test sites that allow for testing significant network sizes (tens or hundreds of nodes), like e.g. Open-Access Research Testbed for Next-Generation Wireless Networks (ORBIT) [10], [11]. However, the access to them is in most cases limited and very costly. As such many researchers tend to evaluate their solutions on the in-house testbed platforms comprising of several laptops or other machines [12]–[15]. The drawback of such an approach is very often that the test suite, due to its characteristic, is limited to very few fixed scenarios [12]–[14], which may not be enough to obtain the wide range of accurate results and is rather useful for a "proof-of-concept" type of experiments. In general, the authors tend to observe that many MANET researchers present their results on the small-scale, simplified models [12], [16]–[18] – both in simulation and emulation environments. Whereas in the case of emulation the limitations may be a result of the hardware capabilities and particular testbed characteristics, for simulations they are usually reflecting the lack of adequate topology and mobility considerations.

In this article the authors present the lessons learned in the course of evaluating the newly proposed IPv6 autoconfiguration solution for MANET networks – the Neighbor Discovery++ (ND++) protocol [19], [20]. The aim was to come up with a high-quality testing approach that would allow for a detailed and accurate evaluation by means of open-source, easily accessible and low-cost methodologies. The authors will depict issues related to the selection of most suitable simulation environment and present the aspects of topology and mobility selection for the reliable simulation set-ups. Referring to the second evaluation stage the simulation-based performance measures will be compared to the real-world testbed experimentation performed at the dedicated wireless testbed platform designed especially to enable creation of multiple MANET scenarios. Both benefits and drawbacks of the two evaluation methods – simulation and emulation are presented, and reveal their complementary nature.

The structure of this article is organized as follows. First the research goal driving the methodology selection is presented in Section 2 and an overview of experimentation objectives and requirements are shown in Section 3. The wireless network simulators for IPv6 experimentation are depicted in Section 4 accompanied with topology considerations in Section 5. Finally Section 6 presents a comparison between simulation- and emulation-based experimentation of IPv6 solutions and a proposed IPv6 testing methodology, whereas Section 7 describes experiences with both of these evaluation methods in IPv6-based MANET networks. Section 8 concludes the article.

2. Research Goal Driving the Methodology Selection

In the course of the authors research the extension to a key IPv6 stateless address autoconfiguration protocol has been proposed [19], [20] – the IPv6 Neighbor Discovery (ND) [21], [22]. The extension – ND++ – is aimed to address the needs of IPv6-based MANET networks and overcome the basic solution limitations, which cannot ensure proper configuration in mobile, ad hoc environments. The proposed ND++ solution has introduced several changes to the basic protocol design depicted in [21], [22]. They are described briefly below, since the character of changes to the common IPv6 stack influences the evaluation methodology. Figure 1 presents the modifications incorporated to the IPv6 packet at different levels. At the ICMP level ND++ uses new message types distinguished by the unique Code field in the ICMP header as well as new option types. There is also a new option proposed for the Hop-by-hop extension header at the IP header level. In addition to these changes, ND++ brings algorithmic modifications of the Neighbor Discovery protocol behavior.

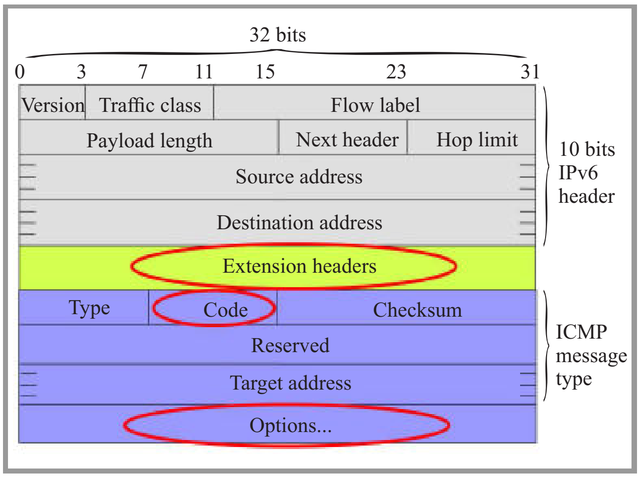

Fig. 1. IPv6 packet overview with marked fields, where changes were introduced by the ND++ protocol.

Considering the characteristic of the ND++ solution presented above, the protocol evaluation and its implementation as a part of an existing IPv6 stack requires code development at IP and ICMP level (mainly in the ND part, but also Extension Headers). Whereas very often dedicated APIs exist for the inclusion of new Extension Header options, changes within ICMP level together with algorithmic changes in the protocol behavior usually require modifications to the IPv6 stack directly. Especially in case of ND++, socket-level programming (used e.g. for many DHCPv6 modifications) cannot be used to implement the whole solution, since basic ND is too tightly coupled with the core IPv6 functionality and cannot be turned on/off or controlled externally. This makes implementation challenging, especially in case of testbed experimentation with the real kernel code modifications.

The proposed ND++ solution evaluation was aiming at improving protocol design and features on one hand as well as evaluating its scalability, performance, and behavior on the other one. To address these needs the protocol evaluation by means of both simulation and real-world testbed emulation was performed.

3. Experimentation Objectives and Requirements

Taking into consideration the particular demands towards experimental evaluation specified by the nature of ND++ research, the authors have identified their objectives and requirements. Thus, the following experimentation environment to be simulated/emulated is envisioned:

- MANET network, where the nodes create ad hoc topologies, there is no centralized server and the network is autoconfigured by means of stateless address autoconfiguration within one network domain; routing is not necessary (including the typical MANET routing protocols), since auto-configuration is performed before routing comes into play during network set-up;
- IPv6 only network – the authors were not interested in dual stack nodes, since ND++ is a purely IPv6 solution;
- Network size from few to hundreds of nodes – a number of nodes depends on the experiment goal; for most simulation scenarios high network node count would be envisioned, however, for testbed-based evaluation an environment with significant network size probably would not be accessible, therefore smaller number of nodes is assumed in this case;

- Possibility to create (simulate/emulate) different topologies – random and pre-selected ones, depending on the particular scenario under test;
- Node mobility can be emulated if necessary.

An exemplary scenario to be simulated and/or emulated, conforming to the environment specified above, is presented in Fig. 2. In this scenario a new node is joining MANET network and performs Duplicate Address Detection (DAD) [20]–[22] as a part of the ND++ based stateless configuration of its newly assigned IPv6 address.

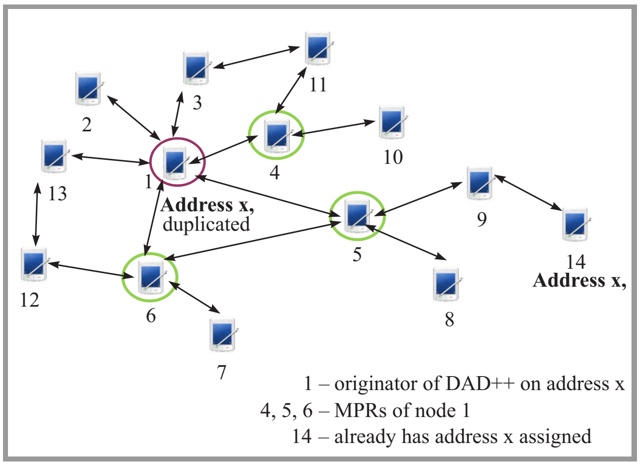

Fig. 2. Exemplary scenario for the ND++ protocol evaluation [20].

Based on the objectives and the research goal defined above a set of requirements towards the experimentation environment is driven. They are particularly specified for testing ND++ solution, however they are representative to most IPv6-based MANET experimentation. Hence, the desired IPv6 MANET test environment should:

- be open-source based;
- have an IPv6 protocol stack incorporated in each node – it must be possible to make modifications to the IPv6 stack implementation directly, since in many cases socket-level changes are not enough; moreover, the IPv6 implementation must be compliant with current Internet Engineering Task Force (IETF) RFCs [23];
- support 802.11 a/b/g network set-ups;
- support large network sizes up to hundreds of nodes (in case of simulators);
- be still actively developed (in case of simulators);
- be used by the research community – this feature ensures obtaining comparative results and their reliability (in case of simulators);
- incorporate visualization tools (not mandatory).

Some of the identified requirements, i.e. the network size, community support and utilization level, refer in fact to the simulation environments – for testbed-based experimentation usually the limitations are imposed with regard to these factors resulting from the availability and access to the hardware experimentation platforms.

4. Wireless Network Simulators for IPv6 Experiments

4.1. Overview

Based on the specified requirements, an overview of the wireless network simulators is presented below, that were investigated as candidate open-source environments for testing IPv6 in MANET networks. The selection contains popular and less-known simulators depicted in MANET-related research papers. Moreover, some simulators for Wireless Sensor Networks (WSNs) were considered as well, since the capabilities of some of them could also be of interest to MANET experimenters. Below the brief overview of the investigated network simulators and their main features is presented:

- NS-2 (Network Simulator 2) – it is a C++ based discrete event simulator [24], which used to be one of the most widely exploited simulators in MANET research. It targets TCP, routing and multicast protocols simulations in wireless and wired networks;
- NS-3 (Network Simulator 3) – it is presented as a discrete-event network simulator for Internet systems [7]. NS-3 is the successor of NS-2 that is gaining an increasing attention of MANET researchers. Similarly to NS-2, it is based on the C++ programming language, however Python API is also available;
- GloMoSim (Global Mobile Information System Simulation Library) – developed by the University of California, Los Angeles, USA (UCLA) [25] using Parsec programming language, became the basis for the commercial simulator QualNet [5]. It used to be popular among the MANET community several years ago;
- OMNET++ – it is an extensible, modular, component based simulation library with several side-projects complementing the core simulator framework [6], [26]. For MANET experimentation especially the INETMANET [6] and OppBSD [27] frameworks are interesting – the first one adds ad hoc functionality and protocols, the second one enables simulations with a FreeBSD operating system (OS) ported to each node, which may be useful in case of the need for an evaluation in a real-world OS. OMNET++ has a wide spectrum of functionality and its popularity has grown significantly in the recent years [6];
- SWANS (Scalable Wireless Ad hoc Network Simulator) – SWANS is a Java-based tool developed at the Cornell University, USA [28]. It leverages the Java

in Simulation Time (JiST) framework [29] to achieve high simulation throughput, good memory utilization and efficient signal propagation computation. Moreover, it allows to run standard Java network applications over simulated networks. Due to its particular concern on simulation performance in terms of resource usage, it enables to simulate large network sizes, exceeding those practically available in NS-2 or GloMoSim [28];

- GTNetS (Georgia Tech Network Simulator) – this tool [30] gathered some attention of MANET researchers several years ago. It became a basis for the MobiREAL project [31], which was proposing a realistic network simulator for MANET networks focusing on the accurate design of mobility models and patterns;
- Sinalgo (Simulator for Network Algorithms) – this software [32], developed by the ETH Zurich, is focused on message exchange, mobility management and topology set-up. These are crucial aspects of MANET networks, however the simulator does not consider inside-node logic and thus does not have TCP/IP stack implementation;
- WSN simulators – the simulators potentially applicable to MANET NETWORKS:
    - AlgoSenSim – framework for simulating distributed algorithms [33], similarly to Sinalgo, it is not protocol stack oriented but algorithm oriented. This framework focuses on network specific algorithms like localization, distributed routing, flooding;
    - NetTopo – designed to test and validate algorithms for WSNs [34], therefore it is algorithm oriented similarly to the previous one;
    - SENSE (Sensor Network Simulator and Emulator) – it has very limited module list, but is interesting due to its emulation capabilities. The simulator focuses mainly on routing in network layer [35];
    - TOSSIM (TinyOS Simulator) – probably one of the most complex WSN simulators. It targets a simulation of Tiny OS nodes and simulates entire TinyOS applications. It works by replacing components with simulation implementations.

4.2. Initial Evaluation

Many of the initially identified MANET simulators depicted above have been discarded at an early selection stage, since some of their features turned out to violate one of the key identified requirements. NS-2 simulator is not developed anymore and is practically superseded by NS-3, therefore the authors have focused on this one in the course of further evaluation. The GloMoSim project finished and the latest release is dated at the year 2000. Since this simulator became the basis for a commercial product, the open-source version has not been further developed since then. There is also a significantly large group of simulators that do not have IPv6 stack – SWANS, GTNetS, Sinalgo, most of the WSN simulators. This is either because it has not been developed so far (SWANS) or as a result of being algorithm oriented (inside node logic is not considered for Sinalgo and WSN simulators like NetTopo and AlgoSenSim). Moreover, SWANS is considered depreciated, since it has not been developed since 2004–2005. There was a SWANS++ effort proposed later on [36], however it reached an Alpha version only, which does not meet our requirements regarding simulator utilization by the community and being under active development. WSN simulators turned out not to be practically useful for simulating IP-level solutions for MANETs, since they are based on the protocol stacks specific for WSNs, not applicable to ad hoc networks. Moreover, they are very often limited to the simulations of only selected networking functionalities, e.g. routing. Interestingly, TOSSIM WSN simulator has its own 6LoWPAN-based IPv6 implementation called Berkeley Low-power IP stack (BLIP). However, as reported in [37], it is currently not completely standards compliant.

4.3. Final Simulator Selection

Having investigated the candidate simulators presented above versus the identified requirements, it turned out that practically only NS-3 and OMNET++ can be an interesting option for testing IPv6 in MANET networks. Hence, the final selection was made between the two of them. Table 1 presents the features of these simulators according to the key requirements.

Table 1
NS-3 and OMNET++ comparison vs. identified requirements

Requirement	NS-3	OMNET++
Can modify IPv6?	Yes	
Implementation up to date?	Yes, but not perfect	Yes, minor bugs
Support for 802.11a/b/g?	Yes	
Support for large network sizes?	Yes, MANET protocols available	
Actively developed?	Very active, support, constant bugfixes	Yes, increasing activity and importance
Other remarks	C++	C++, domain-specific functionality developed as separate projects

Both NS-3 and OMNET++ fulfill the requirements and are capable of making IPv6-based simulations in wireless networks, including MANET-specific protocols. They are currently under constant development with a large support community, which results in high level of their utilization in the research works performed nowadays [6], [7].

The differences influencing the final decision on the simulation environment selection reveal themselves while comparing more detailed simulators features. NS-3 seemed to have better, more bug-free IPv6 implementation. However, it is important to notice that both simulators are actively developed, so the implementations are constantly updated and IPv6 updates are of interest to both NS-3 and OMNET++ teams. Therefore IPv6 code is being improved with each release. There was also a lot of effort put by NS-3 developers to include wide range of accurate mobility and radio propagation models. This enabled to supply it with the capabilities that were often exposed as a weak part of its precursor – NS-2. It is worth noticing, that, apart from outdoor mobility and propagation models, NS-3 contains also an indoor models selection. For these it is possible to position the nodes within the building, specify for it the number of rooms, floors, material from which walls are made, etc. [7]. Moreover, NS-3 is very popular in the community and has a visualization support. As for the OMNET++, its main strength lays in the side-frameworks accompanying the core simulator environment. In the investigated case especially the INETMANET [6] and OppBSD [27] frameworks are of primary importance. INETMANET contains experimental features and protocols dedicated for MANET networks [7]. OppBSD enables to make simulations with FreeBSD ported to each node, which is interesting since FreeBSD is a good target environment for the implementations of IPv6 modifications in the existing kernel code. Unfortunately FreeBSD release ported to the framework is fixed and it is very hard to import the whole release with own modifications. This feature limits practical OppBSD usage, since it is very likely that the FreeBSD version to be used in the real system complementing the simulation work will be different and thus incompatible.

For the purpose of presented research for the evaluation of a new IPv6 solution in MANET networks finally the NS-3 simulator was selected. The most convincing was its strong IPv6 implementation, active developers community and high research community interest which maximizes obtained results credibility.

5. Topology Considerations

The selection of the network topology and mobility patterns properly reflecting the situation in MANET networks is a key aspect in the organization of both simulation and testbed-based experimental evaluation. This issue should be considered on two layers – first one is the position of nodes on the selected area and their interconnections, the second one is the mobility pattern that is applied to such a created scenario. In the mobile ad hoc network mobility influences the topology, so the two are constantly combined together. However, for the evaluation of MANET solutions it is useful to perform part of the evaluation in the static scenarios in order to be able to observe the solution properties before additional factors come into play. The authors will below give an overview of how MANET researchers usually approach topology considerations and propose the strategy that is aiming at maximizing experimentation credibility with regard to this aspect. Referring to the other works the article authors reflect those that present the core MANET protocols evaluation, like e.g. OLSR [38] or OSPF MANET extension [39], or the solutions of a similar nature to the one being investigated.

Topology considerations are treated differently depending on the experiment type. They usually are very limited in case of emulation and testbed evaluation and more sophisticated in case of simulations.

MANET researchers tend to set-up testbed-based experiments with topology generated by hand or from some a priori network settings. As an example, such an approach was applied also in the evaluation of OLSR routing protocol [13]. It enables to obtain proof-of-concept results type with very limited observations possible. Mobility is very often not considered, since in many set-ups (especially in the simple in-house laptop-based testbeds) it is hard to emulate it.

A common approach to the topology creation for the need of simulations is random nodes positioning on the area of a square, a disc or inside a 3D box. Another possibility, similar to the one used for the OSPF MANET evaluation [40], is to place the nodes on a square grid and introduce mobility pattern, e.g. Random Walk, for a specified period of time. The topology “screenshot” after a given time constitutes the node positioning for the experimentation. In the aforementioned OLSR evaluation [13], an automatic scenario generator was proposed to accompany the NS-2 core simulation environment. It allowed for the scenario parameters selection such as mobility, number of nodes, communication parameters, etc. to create a set of random, but in a sense also similar experiments, which could be averaged to obtain final results. With regard to mobility patterns, probably the most popular one for MANET networks is the Random Waypoint (RWP) mobility model. It was used in both OLSR and OSPF MANET evaluation case [13], [40], [41].

NS-3 has a wide range of both position allocators and mobility models available. They not only allow to create the most popular MANET scenarios, but also go a step beyond and create pre-defined indoor scenarios. This, accompanied with realistic radio propagation models, constitutes a powerful tool for MANET research. Moreover, NS-3 allows to provide own topology descriptions created by external topology generators and enables to port the output of the key widely known Internet topology generators – Inet [42], ORBIS [43], Rocketfuel [44] and BRITE [42] – directly to the simulation scripts.

For the extensive simulation experiments it is beneficial to include scenarios based on a realistic, close to real-world topologies. Unfortunately, the topologies generated by the random distributions on a pre-defined areas are not always conforming to the real MANET structures and require high node degree to assure full network connectivity [45]. Therefore several network topology generators can be considered to provide realistic MANET topologies. It would have seemed that the above mentioned Inet, ORBIS, Rocketfuel and BRITE are a good candidate generators. Despite their popularity, they are, however, not suitable for MANET research, since they reflect the Internet fixed network topologies, usually at the autonomous systems level. Moreover, they mostly represent hierarchical router-level topologies, which would correspond to boarder gateways/routers in MANET networks, whereas in MANET research the protocols under investigation usually depict interactions between non-hierarchical MANET nodes being hosts with router functionality inbuilt. An interesting topology generator is the NPART [45], [46]. Although targeted for Wireless Mesh Networks (WMNs), it is based on the assumptions [45] that remain valid also for MANETs and therefore can be applicable to this network type as well. NPART generates topologies similar in nature to WMNs deployed in Berlin and Leipzig. Moreover, it generates connected graph topologies while keeping node degree reasonably low, thus solving an important issue which may occur while using random node placement algorithms. Another solution is the Network Topology Generator (NTG) [47] developed as the module to SciLab open-source numerical computation software [48]. The tool allows not only to generate topologies but also provides a toolchain that enables routing-related analysis on the generated network and provides basic statistics. The authors particularly depict that NTG can be used for MANET simulations. However the tool is in fact oriented towards the design of fixed Internet topologies, whereas MANET simulations are enabled by means of random node distribution accompanied with RWP mobility model. This approach is in fact similar to the standard one described earlier in this section. As such among all the investigated topology generators the NPART tool seams to best fulfill the needs of MANET research.

In order to address the topology considerations in MANET experimentation in a proper manner the authors propose to perform experiments with two different groups of underlying topologies. The first one should be a set of deterministic topologies selected to reflect the particular kind of graphs recognized in graph theory or to expose some particular features of the investigated solution, e.g. circular topology, linear, grid or tree-based topology. The second group consists of random topologies generated in three different manners:

- as the random, most likely uniform, node distribution at a selected area (square, disc, rectangle),
- as the “snapshot” of a topology in the network initiated as a square grid with RWP mobility introduced for a specified amount of time,
- obtained by the network topology generator – the NPART is recommended.

Whereas deterministic topologies allow to verify protocol behavior in some particular, often demanding situations, random topologies are particularly recommended for the simulations aiming at scalability and performance testing with high number of nodes in the network. Generating random topologies by all three methodologies mentioned above ensures maximum accuracy of the obtained results.

For some experiments the network topology selection should be accompanied by the mobility model, which would reflect the MANET environment. Probably the most commonly used by MANET researchers is the aforementioned RWP model [13], [40], [41]. Therefore, this model as a basic scenario is used and recommended. However, it would be beneficial to include also indoor mobility models and more realistic mobility models where the nodes move along the streets, buildings, etc. Unfortunately practical usage of such models in many simulation/emulation environments may not be feasible. There are, however, interesting initiatives like e.g. the MobiREAL project [31], [49] which aims at modeling realistic mobility of humans and automobiles, but their results inclusion into other simulation environments may be in practice impossible.

6. Simulation versus Emulation

Although network simulators are powerful tools enabling simulation of IPv6-based MANET solutions in many diversified environments, there are also several limitations of simulation, which can be effectively addressed by the MANET emulators and testbeds.

Simulators offer the ability to create scenarios with hundreds of nodes in diversified, easily-created network setups. Recreation of such scenarios in the real-world testbeds and emulators is usually impossible, since in many cases the hardware constraints limit them to the few nodes with very limited mobility and/or topology emulation characteristics. However, even if in the emulation-based experiments topology and mobility patterns have to be created artificially and do not result from the real hardware interactions, emulators ensure node behavior closer to reality. This manifests itself especially in the IPv6 experimentation, where emulators based on the real kernel implementations of the IPv6 stack offer better, more bug-free networking stack implementation. At the opposite side the weak point of the currently available simulation platforms is that their IPv6 implementation is not tested thoroughly. As such an unsure behavior may occur, whose underlying source can be hard to identify. This holds true not only for the IPv6 stack implementation, but also for the other elements implementation of the communication stack. The authors will give examples of such a situation in the next section. For emulators, where the real hardware is being used, almost 100% accuracy of bug-free implementation of Wi-Fi drivers and networking protocols can be ensured.

Hence, simulation-based methodology is best suited towards performance and scalability testing, whereas testbed-based experimentation offers not only final verification of the solution in the real-world conditions and a proof-of-concept but also behavior and protocol design evaluation capabilities. These aspects especially reveal themselves in case of testing IPv6 solutions. Therefore, it may be beneficial to exploit emulation-based methodology not only at the testing process final stage, but also at the initial stage, where the emulation environment can help to come up with a well-designed IPv6 solution, capable of operating in the real kernel and integrating with a real hardware. At the next stage such a solution can be evaluated for its performance and scalability – simulation would best address the experiments needs.

Comparing the simulation- and emulation-based experimentation capabilities for IPv6 MANET networks, a testing approach is proposed that would allow for a detailed and accurate evaluation. The goal was to focus on the open-source, easily accessible and possibly low-cost methodologies. Therefore, the usage of both simulation and testbed-based (emulation) techniques interchangeably during the protocol design and evaluation process would be proposed. For the simulation environment an open-source solutions of NS-3 or OMNET++ would be recommended, whereas for the testbed the authors would propose to use possibly high number of machines governed from the central server controlling the experiments and allowing proper topology and mobility emulation. One possible approach to building such a server is depicted in [20] and revealed shortly in Subsection 7.2. The number of minimum 8–10 nodes should allow for obtaining reasonable and quite representative MANET topologies. The more nodes can be afforded, the more diversified results can be achieved. The nodes can run Linux or FreeBSD OS and can be low-cost, even diskless machines equipped with wireless cards for Wi-Fi network set-up. The authors propose to build initial model of the solution in the simulator environment. After tuning the protocol design based on the results obtained with carefully selected topologies and mobility models, following the approach presented in Section 5, testbed trials can be performed which would allow to expose protocol design features that should be improved and would constitute a proof-of-concept. At the last stage the authors would recommend to use the simulation environment with the updated solution description in order to perform scalability testing. In an ideal case it would ease the IPv6 solution development process, if the simulations could be performed by means of virtual machines with real OS representing simulated nodes – such an approach would allow to have only one solution implementation. However, although theoretically possible, such methodology could have practical limitations resulting from difficulties in porting OS versions to the simulation environment.

7. IPv6-based MANET Experimentation – the Authors Experience

In this section the authors will depict their experiences with both – simulation and emulation experimentation of the IPv6 ND++ protocol extension, which was proposed by them for MANET networks. Simulation experiments were conducted in the NS-3 simulator, release 3.11. Emulation experiments were performed on the dedicated hardware testbed platform comprising of 15 nodes based on the FreeBSD operating system in the emulated wireless ad hoc environment.

7.1. Simulating IPv6 with NS-3

Simulations were aiming at the performance and scalability evaluation of the proposed ND++ MANET solution. The implementation in NS-3 environment covered not only the simulation control scripts, but also required to modify IPv6 Neighbor Discovery code of the simulator's IPv6 engine. Therefore a lot of work has been performed with the IPv6 C++ code. In general the implementation was of good quality, as have been expected. However, the authors have noticed that the IETF RFCs, defining IPv6 and its related protocols, were not reflected in detail. Very often basic functionality was ensured, which enabled proper node interactions in most of the standard cases, but more sophisticated features were not implemented or were resolved in a simplified manner. Moreover, throughout the code there were parts of functionality left blank and marked as "to do's" for later NS-3 releases. However, the biggest difference in comparison to the FreeBSD IPv6 networking implementation in the OS kernel was the lack of packet consistency checks. Some packets would have been discarded by the real-world kernel as not conformable to the rules specified in the RFCs, whereas in NS-3 it would be possible that they would have been further processed. Moreover, the researchers have observed that for some issues "own", simplified, solutions have been implemented instead of a very detailed RFC specification. An example of such an approach is the IPv6 address selection procedure in NS-3 release 3.11. However, it has to be underlined that NS-3 is still under active development and in the most current releases IPv6 implementation is being constantly improved. Comparing NS-3 3.11 with the latest release (3.18) several bugs identified previously in the IPv6 implementation were already fixed.

In the performed experiments the NS-3 IPv6-only MANET network was based on the 802.11a/b/g wireless access with nodes creating different topologies (Fig. 3) – pre-defined grid topologies (with and without diagonal connections) and several random node distributions. The investigated scenarios were reflecting a situation, when a new node is joining MANET network and performs ND++ DAD procedure [19], [20] in order to obtain a valid IPv6 address (Fig. 2). In such scenarios some issues with Wi-Fi modules and random numbers at the simulator side affecting simulation experiments were observed. Even for the very basic scenarios, with a grid topology without mobility and small network sizes of less then 30 nodes, the results obtained for different random number seeds were in some cases very distinctive, at the level exceeding expected deviations. After turning off all randomness in proposed solution the issue remained the same and a detailed investigation has shown that the problem is related to the channel access in 802.11-based MAC layer. In the latest release random numbers engine has been significantly improved and underwent major revision, therefore the authors hope that this problem will be solved. Since the experimentation with NS-3 is still a work in progress, the researchers will investigate it further in the course of their future research.

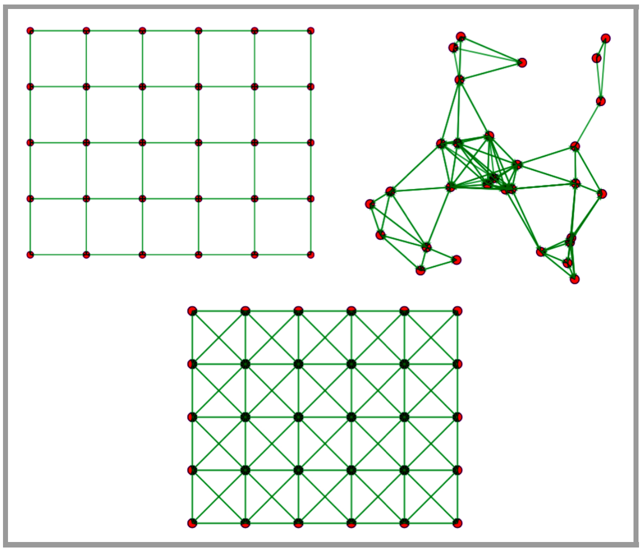

Fig. 3. Selected network topologies used in the NS-3 simulation experiments of ND++.

A consideration of experiences with NS-3 leads to a conclusion that it is more oriented towards TCP-level simulations and one could expect less bugs and issues with such experiments. However, the IPv6 implementation is in general good and would perform well in most of the use cases. Moreover, it is of high community interest to constantly improve the NS-3 simulator, hence the authors would recommend it for IPv6 MANET simulations. An additional benefit of this simulator is also the incorporated tracing and data collection mechanism, which eases processing of diversified result types.

7.2. IPv6 Solution Evaluation on the Testbed Platform

The goal of the testbed-based experimentation was not only to provide proof-of-concept implementation, but also to evaluate the behavior and algorithmic design of the proposed solution. The testbed is a dedicated solution enabling a 15-node MANET network emulation [20]. Its structure is depicted in Fig. 4. Fifteen hardware components connected to the server station and are acting as diskless workstations, which run the kernel version obtained from the server. MANET network is emulated on the Wi-Fi interface of each node by means of an IP firewall (IPFW) packet filtering.

The testbed is based on the FreeBSD operating system with kernel version 7.0. FreeBSD was chosen since it has open IPv6 kernel code implementation (very similar to the one from Linux) with a detailed description of IPv6 networking stack available [50]. One disadvantage of FreeBSD is, however, that updates to the system or its parts are usually not fully compliant among different versions, which causes dependency problems being often hard to handle.

The ND++ solution was implemented as direct modifications to the kernel code. The FreeBSD kernel contains very detailed IPv6 networking stack implementation. As a part of the operating system, it is thoroughly tested and corresponds to RFCs one-to-one with very detailed packet consistency checks and handling all necessary details. This feature is very important while testing protocol behaviors and designs, because it can be assumed that the probability of significant bugs influencing IPv6 protocols behavior is close to zero.

During experimentation none Wi-Fi-level issues have been observed, even though packet-level filtering was used (IPFW rules). At the physical layer all nodes were having direct connection with each other – as such the number of transmissions handled by the Wi-Fi cards in reality was much higher than it was seen at the IPv6 level after filtering. These difficult conditions did not result in problems similar to the ones observed in the simulation environment, which was not that demanding from the physical and MAC layer perspective.

Due to the nature of the ND++ solution, it required modifications to the kernel code directly. Unfortunately socket level mechanism was not enough in this case, however, the ability to use socket-level API would significantly improve the deployment time. When modifying kernel directly kernel recompilation is necessary after each change, which is a very time consuming task. The researchers have managed to reduce recompilation time from over an hour to about 10 minutes, however it still does not ensure comfortable programming. Moreover, bugs not detected during compilation time usually manifest themselves as the kernel panic in a working system. This not only makes debugging process difficult, but also can be hard to handle when the entire operating system crashes. However, the diskless workstations concept introduced in the testbed allowed to deal with such situations easily. Possibly kernel modifications could also be introduced as the kernel modules, which would enable to diminish the recompilation issues.

Although these implementation issues make the code development process difficult, the biggest advantage of the testbed experimentation is that once the solution is implemented and tested the authors can be almost 100% sure about the results and their performance in the real system. Also in the contrary – at the development stage, if something is not working it is almost surely the problem with the modifications, not the kernel implementation. With simulator there is always the risk that some of the models at the simulator core (e.g. Wi-Fi, physical layer, propagation models) were having bugs which may have affected the final results.

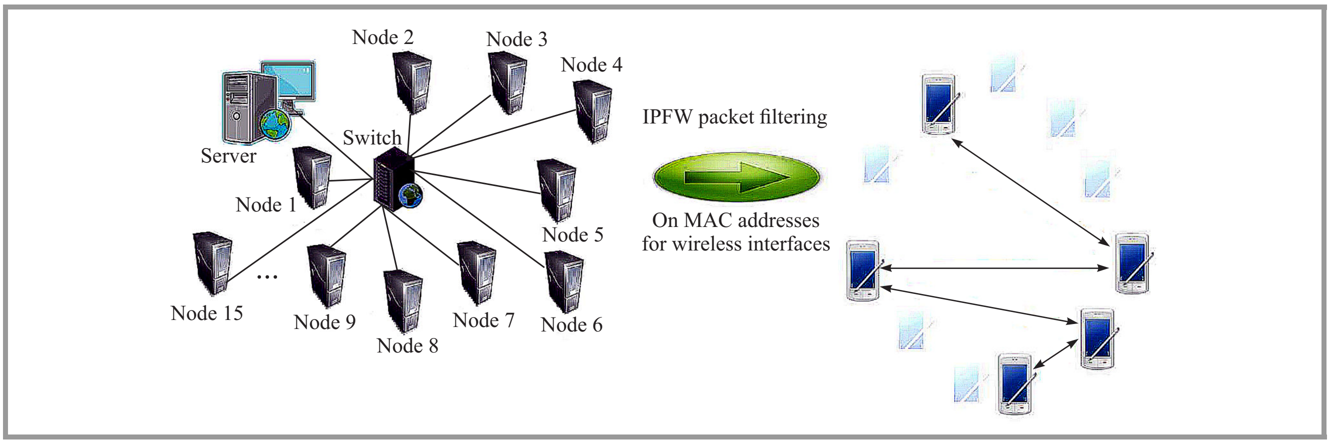

Fig. 4. Overview of the testbed platform set-up [20].

8. Conclusions

With the growing demand towards unsupervised network configuration and operation in the IPv6-based Internet of Things the thorough network testing importance increases. During the course of the authors research the techniques that would provide best test suite for the evaluation of the new IPv6-based solutions in MANET networks were investigated. The findings have shown that relying on simulation or emulation techniques only is not always enough – the variety of complementary techniques is necessary in order to be able to perform both performance and behavioral evaluation. Especially for MANETs, simulation should be complemented by emulation/real-field testing, due to the changing characteristics of such networks and difficulties in reproducing realistic mobility models and network topologies in the artificial environment. The open-source simulation environments investigations reveal, that among quite many MANET simulators only a few can really support IPv6 experimentation. NS-3 simulator best fulfils identified criteria and is probably the most developed and the biggest from the available open-source simulators. This builds its reliability and credibility reflected in the growing attention of MANET researchers. The experiences, though, have shown both the advantages and drawbacks of using this simulator as well as the testbed platform dedicated to IPv6 MANET experiments. As such the authors would formulate the conclusion that simulation and emulation are complementary evaluation methodologies. Therefore, in the proposed testing methodology, accurate testing of IPv6 MANET solutions should exploit both these techniques to maximize credibility and accuracy of the obtained results.

Acknowledgements

This work has been partially supported by EC FP7 EFIPSANS project (INFSO-ICT-215549). Monika Grajzer is a scholarship holder within the project “Scholarship support for Ph.D. students specializing in majors strategic for Wielkopolska’s development”, Sub-measure 8.2.2 Human Capital Operational Programme, co-financed by European Union under the European Social Fund. The authors would like to thank Jacek Serafiński, Marek Tłuczek and Tomasz Żernicki for their help in testbed set-up.

References

[1] P. Stuckmann and R. Zimmermann, “European research on future internet design”, *Wirel. Commun. Mag.*, vol. 16, no. 5, pp. 14–22, 2009.

[2] OECD, “Machine-to-Machine Communications. Connecting Billions of Devices”, OECD Digital Economy Papers, no. 192, OECD Publishing, Jan 2012 [Online]. Available: http://www.oecd-ilibrary.org/science-and-technology/machine-to-machine-communications_5k9gsh2gp043-en

[3] D. Lake, A. Rayes, and M. Morrow, “The Internet of things”, *The Internet Protocol J.*, vol. 15, no. 3, pp. 10–19, 2012.

[4] “OPNET/Riverbed simulator Website” [Online]. Available: http://www.riverbed.com/products-solutions/products/network-performance-management/network-planning-simulation/Network-Simulation.html

[5] “QualNet simulator Website” [Online]. Available: http://web.scalable-networks.com/content/qualnet

[6] “OMNET++ simulator Website.” [Online]. Available: http://www.omnetpp.org

[7] “NS-3 simulator Website” [Online]. Available: http://www.nsnam.org

[8] “InterOperability Laboratory – IPv6 Testing Website” [Online]. Available: https://www.iol.unh.edu/services/testing/ipv6/

[9] “Fraunhofer FOKUS – IPv6 Test Lab” [Online]. Available: http://www.fokus.fraunhofer.de/en/fokus_testbeds/ipv6_test_lab/index.ht%ml

[10] D. Raychaudhuri *et al.*, “Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols”, in *Proc. IEEE Wirel. Commun. Netw. Conf. WCNC 2005 IEEE*, New Orleans, LA, USA, 2005, vol. 3, pp. 1664–1669.

[11] “Open-Access Research Testbed for Next-Generation Wireless Networks (ORBIT)” [Online]. Available: http://www.orbit-lab.org

[12] W. Kiess and M. Mauve, “A survey on real-world implementations of mobile ad-hoc networks”, *Ad Hoc Netw.*, vol. 5, no. 3, pp. 324–339, 2007.

[13] T. Clausen, G. Hansen, L. Christensen, and G. Behrmann, “The optimized link state routing protocol, evaluation through experiments and simulation”, in *IEEE Symp. on Wireless Personal Mobile Communications*, Sept. 2001.

[14] E. Nordstrom, P. Gunningberg, and H. Lundgren, “A testbed and methodology for experimental evaluation of wireless mobile ad hoc networks”, in *Proc. 1st Int. Conf. Testbeds Res. Infrastruc. Develop. Netw. Communit. Tridentcom 2005*, Trento, Italy, 2005, pp. 100–109.

[15] C. Adjih *et al.*, “Experiments with OLSR routing in a MANET”, DTIC Document, Tech. Rep., 2006.

[16] A. Nasipuri, R. Castañeda, and S. R. Das, “Performance of multipath routing for on-demand protocols in mobile ad hoc networks”, *Mob. Netw. Appl.*, vol. 6, no. 4, pp. 339–349, 2001.

[17] S. R. Hussain, S. Saha, and A. Rahman, “SAAMAN: scalable address autoconfiguration in mobile ad hoc networks”, *J. Netw. Syst. Managem.*, vol. 19, no. 3, pp. 394–426, 2011.

[18] S. Boudjit, C. Adjih, A. Laouiti, and P. Muhlethaler, “A duplicate address detection and autoconfiguration mechanism for a single-interface OLSR network”, in *Technologies for Advanced Heterogeneous Networks*, K. Cho and P. Jacquet, Eds. Springer, 2005, pp. 128–142.

[19] M. Grajzer, “ND++ – an extended IPv6 Neighbor Discovery protocol for enhanced duplicate address detection to support stateless address auto-configuration in IPv6 mobile ad hoc networks.” IETF Internet Draft (work in progress), March 2011 [Online]. Available: http://tools.ietf.org/html/draft-grajzer-autoconf-ndpp-00.txt

[20] M. Grajzer, T. Żernicki, and M. Głąbowski, “ND++ – an extended IPv6 Neighbor Discovery protocol for enhanced stateless address autoconfiguration in MANETs”, *Int. J. Commun. Syst.*, 2012.

[21] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, “RFC4861: Neighbor Discovery for IP version 6 (IPv6)”, IETF Draft Standard, Sept. 2007 [Online]. Available: http://www.rfc-editor.org/rfc/rfc4861.txt

[22] S. Thomson, T. Narten, and T. Jinmei, “RFC4862: IPv6 Stateless Address Autoconfiguration”, IETF Draft Standard, Sept. 2007 [Online]. Available: http://www.rfc-editor.org/rfc/rfc4862.txt

[23] “The Internet Engineering Task Force (IETF)” [Online]. Available: http://www.ietf.org

[24] “NS-2 project Website” [Online]. Available: http://www.isi.edu/nsnam/ns/g

[25] “GloMoSim simulator Website” [Online]. Available: http://pcl.cs.ucla.edu/projects/glomosim/

[26] A. Varga and R. Hornig, “An overview of the OMNeT++ simulation environment”, in *Proc. 1st Int. Conf. Simul. Tools Techniq. Commun., Netw. Syst. & Worksh.*, Marseille, France, 2008, p. 60.

[27] “OppBSD framework Website’’ [Online]. Available: https://svn.tm.kit.edu/trac/OppBSD

[28] “SWANS Website” [Online]. Available: http://jist.ece.cornell.edu

[29] R. Barr, Z. J. Haas, and R. van Renesse, “Jist: An efficient approach to simulation using virtual machines”, *Software: Pract. Exper.*, vol. 35, no. 6, pp. 539–576, 2005.

[30] “GTNetS simulator Website” [Online]. Available: http://www.ece.gatech.edu/research/labs/MANIACS/GTNetS/

[31] “MobiREAL project Website” [Online]. Available: http://www.mobireal.net

[32] “Sinalgo simulator Website” [Online]. Available: http://www.disco.ethz.ch/projects/sinalgo/

[33] “AlgoSenSim simulator Website” [Online]. Available: http://tcs.unige.ch/doku.php/code/algosensim/overview

[34] “NetTopo simulator Website” [Online]. Available: http://nettopo.iswsnlab.cn

[35] “SENSE simulator Website” [Online]. Available: http://www.ita.cs.rpi.edu

[36] “SWANS++ simulator Website” [Online]. Available: http://www.aqualab.cs.northwestern.edu/projects/143-swans-extensions-to%-the-scalable-wireless-ad-hoc-network-simulator

[37] “TOSSIM BLIP implementation details – Website” [Online]. Available: http://tinyos.stanford.edu/tinyos-wiki/index.php/BLIP_Tutorial

[38] T. Clausen and P. Jacquet, “RFC3626: Optimized Link State Routing Protocol (OLSR)”, IETF Draft Experimental, Oct. 2003 [Online]. Available: http://www.rfc-editor.org/rfc/rfc3626.txt

[39] R. Ogier and P. Spagnolo, “RFC 5614: Mobile Ad Hoc Network (MANET) Extension of OSPF Using Connected Dominating Set (CDS) Flooding”, IETF Draft Standard, Aug. 2009 [Online]. Available: http://wiki.tools.ietf.org/html/rfc5614

[40] T. Henderson, P. Spagnolo, and G. Pei, “Evaluation of OSPF MANET extensions”, Boeing Techn. Rep. D950, 2005.

[41] P. Jacquet *et al.*, “Performance analysis of OLSR multipoint relay flooding in two ad hoc wireless network models”, INRIA, Tech. Rep., 2002. [Online]. Available: http://hal.inria.fr/docs/00/07/23/27/PDF/RR-4260.pdf

[42] “BRITE – Boston University Representative Internet Topology Generator” [Online]. Available: http://www.cs.bu.edu/brite/index.html#

[43] “Orbis topology generator” [Online]. Available: http://www.sysnet.ucsd.edu/∼pmahadevan/topo_research/topo.html

[44] “Rocketfuel: An ISP Topology Mapping Engine” [Online]. Available: http://research.cs.washington.edu/networking/rocketfuel/

[45] B. Milic and M. Malek, “NPART – node placement algorithm for realistic topologies in wireless multihop network simulation”, in *Proc. 2nd Int. Conf. Simul. Tools Techniq. SIMUTools ’09*. ICST, Brussels, Belgium, 2009, pp. 9:1–9:10 [Online]. Available: http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5669

[46] “NPART – Tool for Realistic Topologies in WMN Simulation” [Online]. Available: http://www.rok.informatik.hu-berlin.de/Members/milic/NPART

[47] “NTG – Network Topology Generator” [Online]. Available: http://atoms.scilab.org/toolboxes/NTG/3.0

[48] “Scilab – Open source software for numerical computation” [Online]. Available: https://www.scilab.org

[49] K. Maeda *et al.*, “Getting urban pedestrian flow from simple observation: Realistic mobility generation in wireless network simulation”, in *Proc. 8th ACM Int. Symp. Model. Analys. Simul. Wirel. Mob. Syst. MSWiM ’05*, Montreal, Canada, 2005, pp. 151–158.

[50] Q. Li, T. Jinmei, and K. Shima, *IPv6 Core Protocols Implementation*. Morgan Kaufmann, 2010.

Monika Grajzer is a Senior Research Scientist and co-owner at Gido Labs, a R&D company. She obtained M.Sc. degree in Telecommunications with honors from Poznań University of Technology, Poland (2008) and is currently pursuing her Ph.D. in the area of IPv6-based stateless address autoconfiguration for mobile ad hoc networks at Poznan University of Technology, Poland. Her research interests are in the area of wireless technologies, in particular in mobile ad hoc networking, network configuration protocols and autonomic, self-managing networking for the Future Internet. She is also interested in the application of these technologies in vehicular communications, home automation systems and low carbon economy solutions. Mrs. Grajzer was working in several European and national collaborative research projects. Her research resulted in the provisional patent application, IETF Internet Draft, several research papers and project deliverables.

E-mail: monika.grajzer@gidolabs.eu
Gido Labs sp. z o.o.
Al. Wielkopolska 29/6
60-603 Poznan, Poland,

Mariusz Głąbowski received the M.Sc., Ph.D. and D.Sc. degrees in Telecommunication from the Poznan University of Technology, Poland, in 1997, 2001, and 2010, respectively. Since 1997 he has been working in the Department of Electronics and Telecommunications, Poznan University of Technology. He is engaged in research and teaching in the area of performance analysis and modeling of multiservice networks and switching systems. Prof. Głąbowski is the author or co-author of 4 books, 7 book chapters and of over 100 papers which have been published in communication journals and presented at national and international conferences. He has refereed articles for many international conferences and magazines, including: IEEE Globecom, IEEE ICC, IEEE HPRS, IEEE Transactions on Communications, IEEE Communications Magazine, Computer Networks, IEEE Communications Letters, IEEE Transactions on Wireless Communications, Performance Evaluation, European Transactions on Telecommunications.

E-mail: mariusz.glabowski@put.poznan.pl
Chair of Communications and Computer Networks
Poznan University of Technology
Polanka st 3
60-965 Poznan, Poland