7th Int. Conference on

Emerging eLearning Technologies and Applications

The High Tatras, Slovakia
November 19-20, 2009

The Semi-automatic Evaluation of Students' Configurations in VIRTLAB Distributed Virtual Networking Laboratory

Petr Grygárek, Zdeněk Filipec, Martin Milata
Department of Computer Science, Faculty of Electrical Engineering and Computer Science
Technical University of Ostrava, 17th listopadu 15, 708 33 Ostrava, Czech Republic
Tel.: (+420) 597 323 243, Fax: (+420) 597 323 099
E-mail: petr.grygarek@vsb.cz, http://www.cs.vsb.cz/grygarek

Abstract. To promote a students' practical skills in computer network courses, we had been using the Virtlab distributed virtual networking laboratory system [1] for a couple of years. As the lab task solutions became a mandatory part of multiple subjects which are attended by hundreds of students, the overhead of manual evaluation of students' configurations worked out on Virtlab system became unacceptable for the teachers. The article presents a technical solution we designed and implemented to automate the checking process and aid teachers with the evaluation of students' work.

Keywords: Virtual laboratory, Distributed learning, Assessment, Collaboration, Course, Education, E-learning, Portal, Teaching, Tools, Verification

1. INTRODUCTION

In technically-oriented courses of computer networking, it is necessary to provide students enough opportunity to practise with real networking devices, which is inevitable to let them gain the skills required from network specialists by the industry. For that reason, well-equipped networking laboratories with sufficient capacity have to be built by teaching institutions and continuously updated to reflect the current state of the technology. Unfortunately, as the number of students grows, more space and lab devices are needed to give students enough chance for individual practising and obtaining the experience with more comprehensive configurations integrating individual technologies into one complete network solution. The another problem is to provide adequate amount of practising to distant-learning students, who physically attend only a very limited number of lab exercises during their study.

Because of the mentioned difficulties, we started a project of "virtual networking laboratory" [2] a few years ago. The original aim of the project that have been called Virtlab was to establish a controlled remote access to networking lab devices which allowed users to reserve a timeslot and utilize networking lab devices 24 hours 7 day a week from any place via Internet [3]. During the following years, we have been creating a fairly complex system which allowed to integrate networking labs of multiple teaching institutions and automatically compose arbitrary network topologies based upon users' requests. The distributed virtual topologies are created from lab devices scattered in multiple sites participating on the distributed Virtlab environment [4], which allows to share lab devices between sites and reduce the overall lab equipment investments considerably. The appropriate lab devices which fulfill user's requirements are found dynamically in all cooperating Virtlab sites. The distributed nature of the resulting semi-virtual environment is completely hidden to the user.

As the original purpose of Virtlab project was to create a teaching environment, meaningful assignments of lab tasks focused on various networking problems are offered to students directly in Virtlab system. In the early years of Virtlab usage, we created mainly monothematic tasks targeted on particular networking protocols and technologies. Students use them individually to enhance skills they obtained in the particular area during regular teacher-led lab work. Later we started to create a more complex task assignments, which teach students how to integrate individual parts into a complete network solution. They are typically used as mandatory projects in different networking subjects and students need to work them out to get the necessary credits.

2. VIRTLAB USAGE IN COMPUTER NETWORKING COURSES

Since the completion of its implementation, Virtlab was used in elective advanced Routed and Switched Networks MSc. course [5] multiple times. In the winter term of academic year 2008/2009 it also became a part of requirements in a mandatory bachelor-level subject Computer Networks, which was attended by more than 350 students at that year. The students' project solved using Virtlab [6] contained 2 consecutive parts. The aim of the first part was to practise IP addressing, VLANs, routing and NAT. In the second part, students practise the configuration of DNS, DHCP and ACLs. That way, student got a chance to get a general picture of a small corporate network design and configuration. Deadlines for submitting of individual project parts were defined in accordance with lecture schedule so that students were motivated to study topics discussed on the lectures during the whole semester.

Individual protocols and networking techniques were also practised on regular teacher-led lab exercises, so that students were able to get a necessary hands-on experience before integrating them into their own project solution. Groups of 4 students worked together on the project. All students worked on the same project, but the assignment were parametrized differently for each group.

From the teacher's point of view, the preparation, handling of assignments and evaluation of students' solutions was rather difficult work. Parameters of assignments for each group had to be maintained in Wiki, separately from the Virtlab system. As the students submitted only paper documentation with devices' configuration and prescribed network operation evidences (such as various command outputs), lot of students just combined the required outputs from solutions worked out by their colleagues, without an adequate understanding of configuration commands and meaning of command outputs. The Virtlab usage statistics revealed that a lot of students even submitted project documentations without accessing the Virtlab system at all, which clearly indicates that they faked the required network operation evidences by creating them using a text editor instead of getting the right outputs from properly configured network devices.

The manual evaluation of documents submitted by students was also very exhausting for teachers, as each group submitted a document with a little bit different structure. The teacher's orientation was further complicated by the fact that the parts of solution which have been submitted consecutively had a different device and interface mapping, as they was developed on independent task reservations in the Virtlab system. The difficulty of the evaluation process caused that we were not able to make groups with less than 4 members, given that we had to deal with 350 students in parallel. The unwanted outcome was that some students didn't contribute to the solution worked out by their group at all. The big groups were also problematic for distant-learning students, who often didn't know each other and had a little chance to meet together physically.

Based on the described bad experience, we decided to design and implement an extension of Virtlab system capable to automate checking of students' configurations and help the teacher with the evaluation of submitted solutions [7]. The main idea was to make students to work out the task in Virtlab system and then apply a set of automatic tests to check the required functionality. That way we completely eliminate a need of creation paper documents by students and their manual evaluation by teachers. Points obtained by students whose solutions successfully passed individual tests are stored directly in Virtlab system and may be exported for further teacher's processing.

3. REQUIREMENTS ON THE VIRTLAB TESTING SUBSYSTEM

Before starting of the Virtlab testing subsystem implementation, we thoroughly analyzed teacher requirements and defined the testing subsystem paradigm so that it will be both flexible and easy to use for teachers and students. We decided to adhere a couple of basic principles, which are shortly discussed bellow.

3.1 Behavior-based testing of students' configuration

As students may take various approaches to configure network devices to reach the required functionality, we decided to test correctness of configurations based on the network response to the testing traffic and checking of internal state of lab devices (i.e. the network behavior) instead on comparing the devices' configuration files with pre-defined solution template. That way we don't need to prepare solution template and formulate task assignment so precisely so that no student may understand it any other way than the author meant it. We also don't need to explicitly define names of user-defined objects used in devices' configurations. The configuration template checking approach would also not be applicable for network devices lacking the possibility of getting their current configuration in text format, which is common for today's network devices managed using Web interface. The only disadvantage of the behavior-based testing approach is that students may try to reach the given behavior with more simpler way than was required (e.g. using static routing instead of configuration of the routing protocol), but it may be rather easily detected using the right testing rules (e.g. checking of OSPF neighbors state). For special cases, when we really need to look into the text-based device configuration, we may send a command to the device to print its current configuration to the console, capture the result and evaluate the configuration correctness using regular expression or Bash script.

3.2 Semi-automatic evaluation with human teacher approval

Although the automatic tests will significantly help the teacher to assess the submitted configuration, there still may arise a lot of special cases which have to be resolved by human teacher manually. The teacher also have also handle situations when some tests could not be performed or they produced an unexpected result. This is why the final decision of how many points will be given to each individual student is made by a teacher, who may adjust points suggested by testing subsystem according to various aspects, not taken into account by the testing engine, that justs sums up the points earned by passing individual tests.

3.3. Compatibility with any kind of networking device

Lab device of any kind may become a subject of configuration testing. The testing subsystem primarily

supposes to work with devices that are controlled using command-line interface, but other kinds of devices' user interfaces (such as SNMP or WWW) may be handled indirectly by testing scripts executed on Linux-based machines incorporated in the lab topology. Various Linux tools (such as snmputils or wget) may be useful for that.

The behavior of the device may be checked either directly or indirectly. Direct tests are performed just on the device where the required functionality had to be configured. They include checking of response on status commands, outputs from debug messages and listing of device configuration. On the other hand, indirect tests are performed from different device than the one in question. Indirect tests evaluate the response to the probing traffic, which may be either processed, silently passed, modified or filtered by the tested device. The typical usages of indirect tests include checking of ACLs, NAT or firewall configuration. They are also used for testing of non-CLI managed devices, as mentioned above.

3.4 Group-based task solution and group-specific assignment parametrization

To promote student's ability of team work, we considered important to let students to solve their tasks in groups. The testing subsystem allows to create groups of any number of students. All members of the group are treated as equal, i.e. every group member may submit a solution or change a solution previously submitted by other group member, provided that the period in which students are expected to submit solutions of specific task or its part is not over.
Each group may be given a specific part of a task assignment, which parametrizes a general assignment and will be taken into account during tests of the submitted solution.

3.5 Flexibility and ease of usage

The testing subsystem design has to take into account that teachers may want to state requirements for submitting tasks in many different ways. Solutions of tasks may be either submitted at once or in consecutive parts, which allows students to get teacher's feedback on the submitted part before advancing to the following part. If students need to make decisions which considerably influence the resulting solution (e.g. subnetting of a given address range or selection of the device to host particular network service), there must exist a mechanism how to enable students to describe their decisions in a structured way defined by the teacher. The testing subsystem may then utilize the information obtained from students during evaluation of the particular students' configuration. Teacher may see these students' inputs as well.

Regardless of the potential complexity of definition of tests and parametrizations for multiple student groups, the user interface has to be easily understandable for students who don't know anything about the testing subsystem and just want to work out the configuration and submit it after they are happy with the operation of their solution.

4. THE TESTING SUBSYSTEM DESIGN AND IMPLEMENTATION

The steps which the teacher and students follow while using the testing subsystem are shown on Figure 1. To automate evaluation of submitted students' configurations, teacher first has to define tests, create appropriate testing rules and provide task parametrizations for each student group. Each test is defined by a source device from which it will be performed, the testing rule specifying the testing method, binding of testing rule variables to parameters assigned to student group and inputs obtained from the user and finally the number of points the student may earn when the particular test is passed successfully.

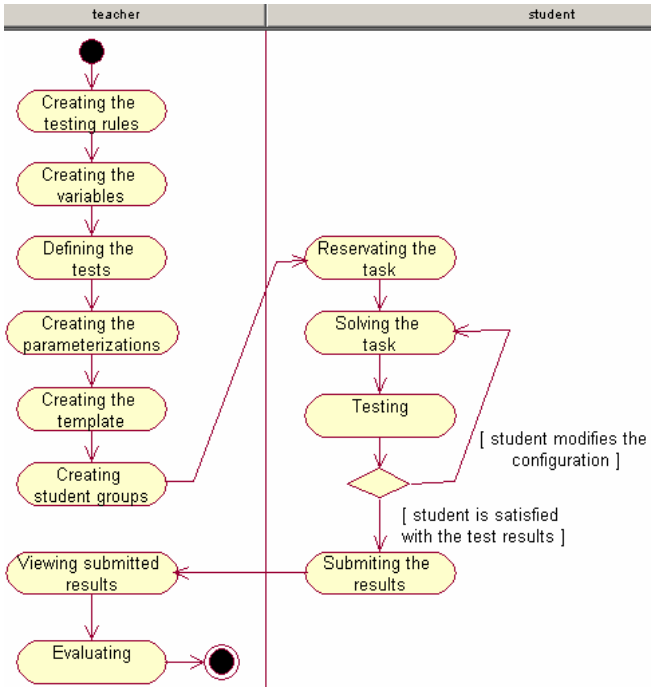

Fig. 1. Steps of testing subsystem usage

The testing rule describes how to perform each test. Every testing rule consists of the rule script, response end marker and validation regular expression. The rule script is the sequence of commands which will be sent to the test source device to obtain a particular response which will be further analyzed. Variables may be used in rule scripts. The variable names used in the test script will be replaced by values of appropriate user or parameterization variables during the test execution, as will be explained later. The validation regular expression is used to validate the source device response generated after completion of testing script. As the output generated by source device may be potentially long, it is necessary to delimit a part of source device response to be evaluated by validation expression. The response end marker is used for that purpose.

There are two kinds of the end markers - regular expression end marker and time-based end marker. Once the script is sent to the source device, response of the device is being read until the received output matches the regular expression end marker or the time specified by the time end marker elapses. After the necessary part of the source device output is read, it will be evaluated by validation regular expression.

In the second step the teacher defines so called parameterization and user variables, whose values will be bound to testing scripts variables during execution of each test. Each variable is defined by it's name and the purpose description which will be displayed to the user. There exist two kinds of variables – paramerization variables and user variables. Values of parametrization variables are taken from assignment parametrizations defined for each student group, whereas user variables are requested from the user when he or she requests to test the completed solution. The usage of variables in the testing process is depicted on figure 2.

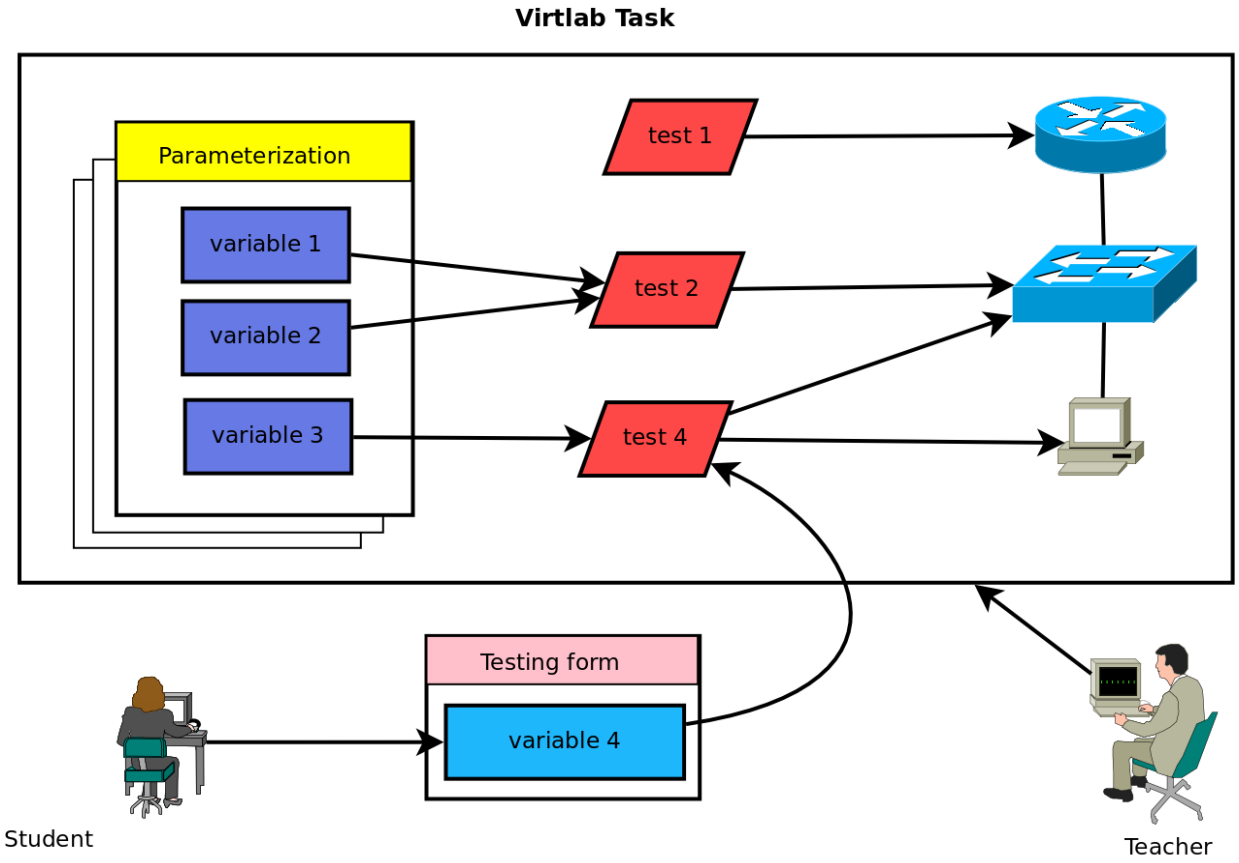

Fig. 2. Usage of parametrizations and user variables during execution of tests

After at least one testing rule us defined, the teacher may start to create tests for a particular task. The test is defined by one testing rule and bindings of script variables to parametrization and user variables, if variables were used in testing rule script. Furthermore, the teacher has to specify on which source device the test should be executed, how many points the students will get if the test is passed successfully and whether it is compulsory to pass the test successfully. One testing rule may be used to create one or more tests. The tests created from the same testing rule can be executed on various source devices and with different bindings of variables. The relationship between testing rule, test and variables is shown on figure 3.

If the created testing rules reference parametrization values, the teacher needs to create an assignment parametrization for every student group. Every parameterization consists of a set of values of parameterization variables and an optional group-specific part of the task assignment text. Refer to figure 3 to see how parameterization variable values are used in the testing process.

As mentioned above, the testing subsystem allows the teacher to require students to solve a task as a series of consecutive parts. For that purpose the teacher has to define the template which describes parts of a complete task solution. Every solution part is specified by the time period during which it has to be solved and submitted. The teacher also chooses which tests will be performed on every solution part and optional extra information and files the students have to enclose when submitting each solution part. Figure 4 shows an example of a template with two solutions parts.

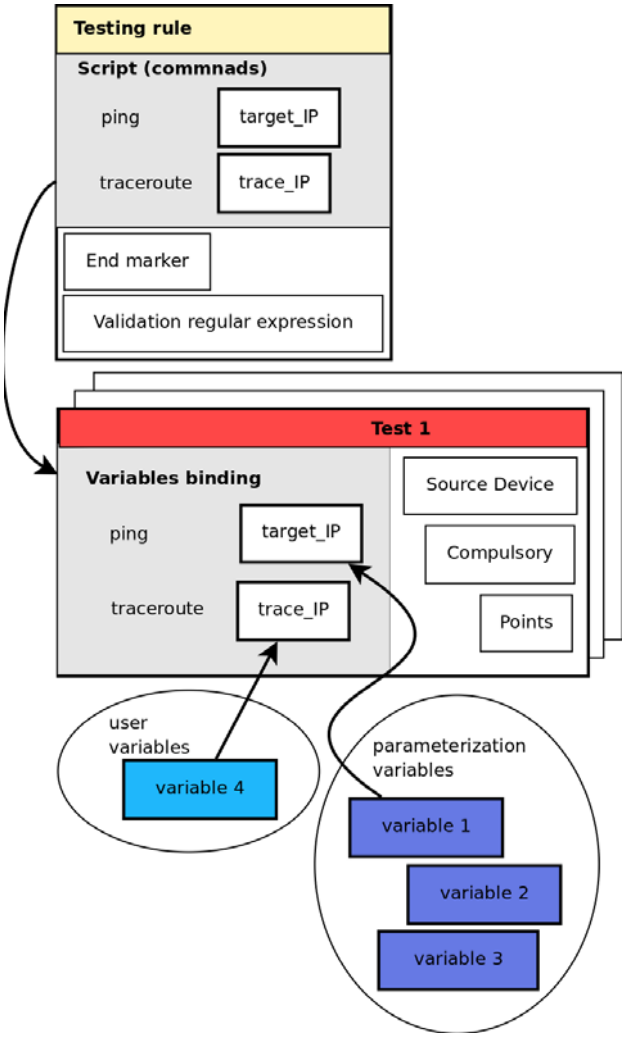

Fig. 3. Relationship between testing rule, test and variables

The last step to be done before the students can start to work on particular task's solution parts is to create student groups and to assign a parameterization to each group. Once the student becomes a member of the student group, he or she can reserve task in Virtlab system and solve it according to the general task assignment and parameters assigned to his or her group. After completion of the configuration the student can ask the system to execute tests defined by a teacher for the solution part. The parametrization assigned to a given user will be used as source of values of parametrization variables bound to variables used in a testing script. If there are any user variables used in testing script, the system will ask the student to fill so called testing form with values of required user variables first.

After all tests are done, the system informs the student about the individual test results. An automatically generated report informs the student how many points he or she got based on successfully passing each individual test. In case of student satisfaction with the test results he or she can fill the submit form, enclose all required attachments and auxiliary information, and submit the task solution part. If the student is not satisfied with the test results, he or she can repair the configuration and ask the system to do a new testing. The process may be repeated as many times as the student wants. The testing subsystem allows the student to save the previously entered values of user variables to avoid a need to fill the testing form every time the student wants to execute tests. The saved variable values are visible and shared by all members of the same student group.

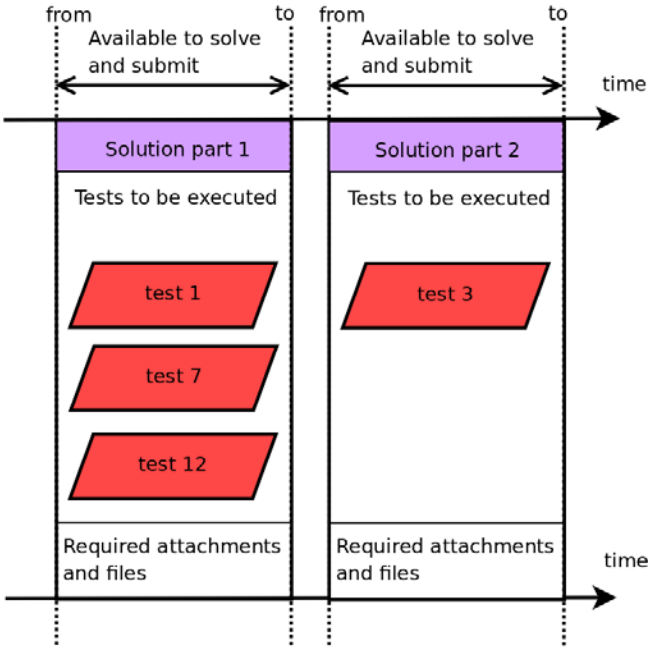

Fig. 4. Template of solution parts

The teacher can browse through the submitted solutions and test results at any time. After the time period for completion the solution part is over, the teacher can enter the final evaluation based on the test results and attachments for each solution part. After the end of the semester, the teacher may archive results of all students. The archived information is kept aggregated in separated data storage.

5. CONCLUSION

The described semi-automatic configuration checking subsystem is planned to be used for the very first time in the winter term of academic year 2009/2010. We believe it will help teachers with student's configuration evaluation considerably and make the whole process much more objective and predictable for students. We plan to monitor the students' attitudes and opinions of the new learning method carefully and describe our results in a subsequent conference papers.

6. REFERENCES

[1] Grygárek,P., Milata, M., Vavříček, J.: The Fully Distributed Architecture of Virtual Network Laboratory.

In proceedings of ICETA, Stara Lesna, High Tatras, Slovakia, 2007. ISBN 978-80-8086-061-5

[2] Němec, P.: Virtual Network Laboratory. Master's Thesis, Faculty of Electrical Engineering and Computer Science, VŠB-TU Ostrava, 2005 [In Czech]

[3] Grygárek, P., Seidl, D., Němec, P.: Enabling Access to Equipment of Computer Network Laboratory for Practical Trainging via the Internet. Proceedings of Technologies for E-Learning conference, FEL ČVUT Praha, 2005, ISBN 80-01-03274-4, pp. 43-52. [In Czech]

[4] Grygárek,P., Milata, M.: Piloting Environment of Distributed Virtual Networking Laboratory. Proceedings of Virtual University, Bratislava, Slovakia, 2007, ISBN 978-80-89316-09-0, pp. 209-212.

[5] Grygárek,P.: Routed and switched networks [online]. [2008] [cit. 2009-06-29]. Available at WWW: http://www.cs.vsb.cz/grygarek/SPS/SPS-pozadavky-dalkari-0809.html

[6] Grygárek,P.: Assignment of Student's projects [online]. [2008] [cit. 2009-06-29]. Available at WWW: http://www.cs.vsb.cz/grygarek/POS/projekt0809Z.html

[7] Filipec, Z.: Automated evaluation of configurations in distributed virtual network laboratory. Master's Thesis, Faculty of Electrical Engineering and Computer Science, VŠB-TU Ostrava, 2009 [In Czech]

THE AUTHOR(S)

Petr Grygárek (Ph.D, MSc.) is a professor-assistant at Department of Computer Science at VŠB-TU Ostrava. His professional interest is focused on computer networking, distributed systems and computer hardware. He is a coordinator and instructor of Regional Cisco Networking Academy and coordinator of Virtlab development group. He holds CCNP, CCNA and Network Security courses teacher's certificate.

Zdeněk Filipec (MSc.) just graduated at Faculty of Electrical Engineering and Computer Science of VŠB-TU Ostrava and is willing to continue in his doctoral study. The goal of his master thesis was to analyse, design and implement the testing subsystem described in this article. He is a CCNA-level instructor of Regional Cisco Networking Academy.

Martin Milata (MSc.) is a Ph.D. student at Department of Computer Science. His thesis is focused on problems of routing in mobile ad hoc networks. He is interested in computer networks and hardware. He is instructor of Regional Cisco Networking Academy (CCNA-level).