SIMULATION OF AGENT TEAMS: APPLICATION OF A DOMAIN-INDEPENDENT FRAMEWORK TO COMPUTER NETWORK SECURITY

Igor Kotenko
Computer Security Research Group
St. Petersburg Institute for Informatics and Automation
39, 14th Liniya, St. Petersburg, 199178, Russia
E-mail: ivkote@iias.spb.ru

KEYWORDS
Agent-based Simulation, Intelligent Agent Teams, Network Security, Distributed Denial of Service.

ABSTRACT

The paper proposes an approach for multi-agent simulation of intelligent agent teams' collaboration and competition. This approach is for representing the complex processes ongoing in different subject domains by using various types of agent teams and their interactions. Teams can be in relations of indifference, antagonistic and non-antagonistic competing or (and) various kinds of cooperation, and may adapt to the actions of opponents or collaborators. The domain-independent multi-agent framework and common simulation environment are proposed. The suggested approach differs from other related approaches by attempting to formalize main components needed for agent teamwork and team collaboration and competition. The approach is applicable for the tasks of competitive activity in commerce, counteraction in the Internet, simulation of rescue operations, coalition actions, military conflicts, etc. We consider the theoretical models, implementation and investigations of the proposed approach on an example of simulation of distributed Internet attacks and defense.

1. INTRODUCTION

Due to a set of reasons the distributed complex systems cannot be automated effectively on the basis of traditional architectures, methods and software tools. These systems demand the possibility of implementing the dynamic behavior, autonomy and adaptation, using the methods based on negotiations and cooperation that lie in the basis of agent-oriented systems. The effective implementation of these systems requires thorough and comprehensive simulation.

There was developed the variety of frameworks and architectures for multi-agent simulation of distributed complex systems, e.g. shared plans theory (Grosz and Kraus 1996), joint intentions theory (Cohen and Levesque 1991), hybrid approach (Tambe 1997), there were implemented various software multi-agent environments (Macal and North 2005; Marietto et al. 2002). The approaches based on belief-desire-intention (BDI), distributed constraint optimization (DCOP), distributed POMDPs, and auctions or game-theoretic (Tambe et al. 2005) are emphasized. Different mechanisms for collaborative agent team maintenance are suggested (Kaminka et al. 2007). The main task of these approaches is *to provide the optimal interaction of heterogeneous components (including robots, smart devices, machines, software agents and people) to reach some high level goal*. These methods, models and tools have been applied in different subject domains (Haque et al. 2005; Jennings 1995; Kaminka and Frenkel 2005; Tambe 1997; etc.). But the approaches mentioned lack the detailed description of the aspects needed for simulation of competing agent teams. There are several multi-agent simulation system implementations for imitating antagonistic opponents. For example, Robocup Soccer implementations use mainly reactive agents, TacAir-Soar simulates combat pilots, there is a network intrusion simulation implementation for MASON simulation toolkit (Luke et al. 2004), etc.

The paper proposes an *approach that allows to simulate and investigate the work of collaborating and competing agent teams*. We examine the interaction of at least two agents teams that can be in the relation of antagonistic and non-antagonistic competing or (and) various kinds of cooperation, and may adapt to the actions of opponents or collaborators. The approach suggested in the paper differs from other related approaches by (1) attempting to formalize main components needed for agent teamwork and team collaboration and competition, (2) emphasizing various types of interactions between agent teams and (3) developing the multi-level simulation environment that allows to use as the basis the various discrete-event simulators and the libraries (ontologies) of different subject domains. *The global goal of our research is* to develop domain-independent framework and common simulation environment (integrated software tool) for analysis of intelligent agent teams as well as to implement them for investigation of several important application domains: simulation of rescue operations, simulations of social-technical networks, simulation of Internet attacks and defense, etc.

The paper is organized as follows. *Section 2* outlines the proposed approach to the simulation of competing agent teams. *Section 3* describes the architecture of proposed simulation environment. *Section 4* considers

Proceedings 23rd European Conference on Modelling and Simulation ©ECMS Javier Otamendi, Andrzej Bargiela, José Luis Montes, Luis Miguel Doncel Pedrera (Editors)
ISBN: 978-0-9553018-8-9 / ISBN: 978-0-9553018-9-6 (CD)

several important domains which require the investigation of competing agent teams' behavior. *Section 5* examines the application of proposed approach for the multi-agent simulation of counteraction of malefactors and defense systems in the Internet. *Section 6* represents the conducted experiments. *Conclusion* surveys work results and future research.

2. SIMULATION FRAMEWORK

The proposed approach supposes that counteraction, competition or cooperation in a particular system (subject domain) is represented as the interaction of various teams of software agents. The aggregated behavior of this system is revealed due to local interactions of particular agents in the dynamic environment defined by the agent interaction model. Classes of agent teams are selected depending on the solved task and subject domain. For example, for some tasks of information counteraction one can use at least the following three classes of agent teams according to the realized strategy: the class of attack teams; the class of defense teams; the class of agents imitating some background processes. In this case the goal of attack team is to detect the vulnerabilities of the environment and defense system and to realize the given list of threats due to executing the attack actions. The goal of defense team is to protect the environment and its own components from these actions. Agents of different teams compete to reach the opposite goals, collaborate to fulfill the attack or defense task or are in an indifference relation. Agents of the same team collaborate to reach the joint intention (to realize the threat or to defense the environment).

The developed *domain-independent multi-agent framework for simulation of agent competing* includes the following components: (1) The ontology of application domain containing application notions and relations between them (we differentiate the problem ontology, the shared application ontology, the application ontology of particular team and particular agent); (2) The protocols of teamwork for the agents of different teams; (3) The models of scenario behavior of agents for team, group and individual levels; (4) The libraries of agent basic functions; (5) The communication platform and components for agent message exchange; (6) The models of functioning environment, including topological, functional and other components; (7) The models that provide the interaction of teams (antagonistic and non-antagonistic competing or various kinds of cooperation). The approach for teamwork is based on combining the elements of joint intentions theory, shared plans theory and combined approach. The mechanisms of agent interaction and coordination are based on the procedures of action consistency maintenance, monitoring and agent functionality recovery, and communication selectivity (Tambe 1997).

The following main *simulation components* are represented on the basis of this approach: models of agent teams; models of team interactions; interaction environment model. *Models of agent teams* are intended for the representation of investigated processes. They include particular team ontologies, agent basic functions, agent classes, agent protocols, behaviour scenarios. *Team ontologies* are based on the subject domain ontology and include the notions and relations used by agents of this team. The list of *agent basic functions* includes the following functions: initialization; shutdown; access to the agent ontology; management of active agents list; basic work with transport-level modules (connection establishing, message sending, connection closing). The needed *agent classes* are defined for the teams. The amount of agents of predefined classes is set in each team. *Agent interaction protocols* are represented as the sequence of instructions with specific parameters. The type of instruction defines how to use these parameters. The conditions of protocol initialization provide communication selectivity for agents. Agent interaction protocols are based on the transport layer that is provided by the communication environment. For example, the developed protocol for agent team establishing is based on dividing agents into "clients" and "servers". The first send the messages about their existence to "server". Server contains the list of agents in the team. Periodically it checks the agents in this list to actualize it and to know which of agents are active. Agent team establishing protocol is the part of procedures for monitoring and recovery of agent functionality. *Scenarios* represent various stages of team actions. Adaptation procedures are implemented in scenarios to act depending on other team actions and environment reaction. Agent teams' behavior scenarios ensure action consistency maintenance. *Models of team interactions* include the models of antagonistic competing, team cooperation and adaptation. *Model of antagonistic competing* lies in the basis of competing teams' interaction. This model defines the goals, subgoals, intentions and actions of competing teams that are aimed on the interaction environment or (and) the opponent team. *Cooperative interaction* happens between teams that pursue the same goal. The proposed model of cooperation is based on the exchange of information between teams. Such exchange is made to raise the effectiveness of reaching the common goal and occurs on several different levels with the use of agents of various classes. For example, in the task of cooperative network defense simulation it is possible to exchange attack signatures, network traffic data, filtering requests, etc.

Adaptation is in reacting to the actions of other teams and environment changes by modifying the scenarios of team behavior. *Adaptation model of agent team* T_1 according to the actions of competing team T_2 and environment changes is in the following. When the efficiency (or severity) $S(t)$ of team T_2 actions and (or)

the interaction environment changes, then such configurations and scenarios $K(t)$ of team T_1 behavior are chosen, that ensure the reaching of the given goal with minimization of sum of action cost functions C_i, i=1,…, n (n – number of main agent team goals):

$$F = \sum_{i=1}^{n} C_i\left(S(t), K(t)\right). \qquad (1)$$

Model of environment for agent teams' interaction allows determining such interaction environments that are characterized by various representation granularity that depends on the requirements for simulation fidelity and scalability.

3. SIMULATION ENVIRONMENT

To implement the proposed approach the special simulation environment (tool) was developed. To implement the approach the multilevel architecture of simulation environment was proposed. It consists of the following layers (Figure 1). *Simulation Framework* is supposed to be discrete-event simulator. Other simulation environment components are Simulation Framework modules and models. It can use for its functioning the various domain-oriented discrete-event simulation software tools and software libraries (Network Simulators; Simulation frameworks; Simtools; etc.). *Environment Simulation Framework* is a suite of simulation modules that allows to imitate realistically the environment for agent interaction. This component implements the communication environment and transport protocols models. Multi-agent simulation is realized due to *Agent-based Framework* that uses Subject Domain Library. The framework is the modules library that defines intelligent agents implemented as applications. FIPA (FIPA) abstract architecture elements are used during agent modules design and implementation. Agent communication language is used to allow the agents interact with each other. The messages between agents are transmitted on the top of transport protocol that is implemented in Environment Simulation Framework. Agent directory is obligatory only for the agent that coordinates the actions of other teammates. Agents are able to control other modules due to messages. *Subject Domain Library* (ontology) is the library that contains modules for imitation of subject domain processes. The libraries for different domains are supposed to be implemented and used.

The parameters of agent teams, their interaction and interaction environment are used as the input for the simulation tool. Simulation of the studied processes is made on the basis of the mentioned models. Simulation tool outputs the set of various parameters using which one can analyze the simulated system. Simulation process is defined due to the technique that includes various stages for parameters' input and processing, simulation of agent teams' behavior, parameters output and analysis.

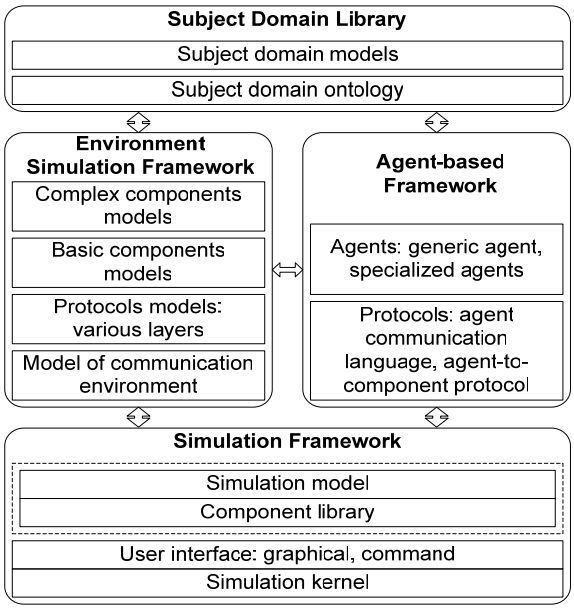

Figure 1. Simulation environment (tool) architecture

4. EXAMPLES OF APPLICATIONS

We are intending to investigate the application of the agent-based framework and software environment for simulation of agent teams in several important domains and develop application-dependent models and software tools. Below three different applications are considered which are under investigation.

Simulation of rescue operations. In RoboCup Rescue, agents work in the virtual city. For example, agent teams can include a human fire chief and a number of robot or agent-controlled and managed fire brigades. The human fire chief is given a high-level view of the fire-fighting progress and can make better role-allocation decisions to save trapped and injured civilians and limit damage. The tasks are to rescue the civilians, buildings which are on fire and roads that are blocked. The problem is then how to assign ambulance teams, fire brigades and police forces to these tasks. We propose to represent several rescuers (e.g. fire brigades, ambulance and police) as an agent teams. These teams pursue the same goal and are able to cooperate and to adapt to each other actions. The cooperation is in asking the other team for the corresponding agent for help, e.g. when additional fire brigade is needed. Each rescue team adapt to the actions of another team by monitoring the current rescue operation and if the other team manages the situation than the first team switches to another operation. There can be existed additional teams representing the opposite site: the terrorists that are the reason of rescuing operations. They try to act in such a way that rescue operations will fail or be ineffective. This can be done to simulate the "worse case scenario". Also it is very important to simulate the ordinary people actions. They can be represented as a team of agents that needed to be rescued if they are injured.

Simulations of social-technical networks. There can be a lot of applications of the proposed approach in social networks (Rosvall Sneppen 2008). For example, stock market game, epidemic propagation, rumor spreading, etc. The proposed approach can be widened to represent a big variety of social relations and interactions based on them (excluding the already stated indifference, counteraction, cooperation and adaptation): values, ideas, financial exchange, friends, kinship, dislike, trade, web links, disease transmission (epidemiology), etc. These new interaction types also might be used to simulate other subject domains with higher fidelity. Simulating the antiterrorism actions is the most natural usage of the approach proposed. The simulated "terrorists – antiterrorists" processes are represented as counteracting teams of agents. They have the opposite goals. The teams from the same site cooperate and adapt to each other and opponents' actions. There is an important issue of cooperation between heterogeneous agents as humans and machines, or e.g. human squad and helicopter squad. It is proposed to create models of interaction on the different layers. Simulation of the competing companies is one of the other important applications. These companies are represented as the teams of competing agents. They interact not only on the basis of competing, cooperation and adaptation. There are financial exchange, trade and other types of interactions that have to be specified.

Simulation of distributed Internet attacks and defense. The investigation of counteraction of malefactors and defense systems in the Internet is one of the important tasks that have to be solved for designing the secure distributed computer systems. The design and implementation of effective cyber-defense intelligent cyber-defense system is a very complicated problem. According to our view the prospective network cyber-defense system has to be *fully integrated and multi-echeloned* one. To effectively detect the computer attacks or unauthorized operations and to flexibly react on them, it is needed to carry out the continuous control of network functioning, analyze possible risks, collect knowledge about counteraction, detection and reaction methods and use them for defense reinforcement. Besides, the effective cyber-defense should include the mechanisms of attack prevention, detection, source tracing and protection as well as can only be achieved by the *cooperation of different distributed components* which can be represented as different agent teams. This approach was implemented for the investigation of distributed denial of service attacks and defense mechanisms (Kotenko and Ulanov 2006; Kotenko and Ulanov 2007). Let us consider the last application domain in more details.

5. COMPETING AGENTS IN INTERNET

The investigation of counteraction of malefactors and defense systems in the Internet is one of the important tasks that have to be solved for designing the secure distributed computer systems. The representation of malefactors and defense systems as intelligent attack and defense agent teams are a very prominent idea. It is very topical in viewing the current tendencies in malware technologies which will emerge from the combination of classic network worms and viruses, distributed denial of service (DDoS) tools, root and kernel kits, botnets as well as concepts from network and artificial intelligence research, including peer-to-peer (P2P) communications and intelligent agents. In the Internet security context, the defense of traffic that is generated by the agents-users is one of the important tasks for the defense agent teams. This is vital, for example, during counteracting the teams of agent-malefactors that realize DDoS attacks.

Traditional defense from *DDoS* includes detection and reaction mechanisms. To detect abnormal network characteristics, many methods can be applied (for instance, statistical, cumulative sum, pattern matching, etc). The examples of detection methods are Hop counts Filtering (HCF), Source IP address monitoring (SIPM), Bit per Second (BPS), etc. As a rule, the reaction mechanisms include filtering, congestion control and traceback. As detection of DDoS is most accurate when it is close to the victim hosts and separation of legitimate is most successful close to the sources, adequate victim protection to constrain attack traffic can only be achieved by *cooperation of different distributed components* (Mirkovic et al. 2005). There are a lot of architectures for distributed cooperative defense mechanisms, e.g. Server Roaming, Market-based Service Quality Differentiation (MbSQD) (Mankins et al. 2001), Transport-aware IP router architecture (tIP) (Wang and Shin 2003), Secure Overlay Services (SOS) (Keromytis et al. 2003), ACC pushback, COSSACK (Papadopoulos et al. 2003), Perimeter-based DDoS defense (Chen and Song 2005), DefCOM (Mirkovic et al. 2005), Gateway-based (Xuan et al. 2002). During DDoS attack and defense counteraction, the agent teams realize distributed action scenarios and adapt to each other actions. To realize DDoS attack, malefactors have to do the following: compromise a lot of hosts, deploy on them attack agents and then make a simultaneous attack on victim hosts or the entire Internet subnets. In turn, defense agents that try to block the traffic generated by the attack agents have to provide normal traffic transmission. The team of agents-malefactors evolves due to generating new attack types and instances, and the scenarios of their realization to overcome the defense. The team of defense agents adapts to the malefactors actions due to changing the security policy, choosing, creating and using new instances of defense mechanisms and profiles.

The proposed simulation approach was implemented to investigate the behavior of attack and security agent teams. The developed *attack team ontology* includes the notions and relations of subject domain "DDoS attacks". Attack team model contains two agent classes:

"daemon" executing attacks and "master" coordinating other attack agents. Attack team uses the following protocols: team establishing; workability check; attack parameters transferring; attack realization. The scenarios developed define the attack agent team sequence of actions. *The defense team ontology* includes the notions and relations of subject domain "Defense mechanisms against DDoS". The following classes of defense agents are proposed: initial data processing ("sampler"); attack detection ("detector"); traffic filtering ("filter"); traffic limiting ("limiter"); agent of investigation. Defense team uses the following protocols: team establishing; sampler data collection; sending the addresses of possible attack sources; attack agents defeating, etc. Scenarios define team's action sequences. The agents "limiter" apply various modes. The following mechanisms are used by detector: Hop Count Filtering (HCF), Source IP Address Monitoring (SIPM), Bit Per Second Filtering (BPS).

The model of antagonistic counteraction of attack and defense agents defines the following aspects of attack realization. Attack team during attack influences on the network and attack target and this impact is transmitted to the defense team via the network. Defense team during attack detection tries to apply some countermeasures due to the defined defense mechanisms: traffic limiting, traffic filtering and attack agents defeating. Defense team influences on the network by traffic limiting and filtering. It influences on attack team also: particular agents might fail to receive attack commands and the attack packets might be filtered also. Trying to defeat attack agents the defense team influence on them directly.

Cooperative interaction happens between teams that aim the common goal to defend or to attack the network. Defense teams exchange information to raise the effectiveness of attack counteraction, and attack teams do the same to make the attack more effective. The following four agent classes are used in the proposed approach for cooperation: samplers, detectors, filters and agents of investigation. Defense teams can interact due to various *cooperation schemas*: *no cooperation* – all teams work on their own; *filter-level cooperation* – defense teams can apply filtering rules on the other teams filters; *sampler-level cooperation* – defense teams can receive traffic data from the other teams samplers; *weak cooperation* – defense teams can use some filters and samplers from other teams; *full cooperation* – defense teams can cooperate using all available data and filtering rules.

According to the proposed *adaptation model (see formula (1))*, when attack traffic volume $S(t)$ changes the defense team chooses such team configuration and behavior $K_D(t)$ that minimize the amount of false positives $C_{FP}(S(t),K_D(t))$, false negatives $C_{FN}(S(t),K_D(t))$ and attack duration $C_T(S(t),K_D(t))$:

$$\min_{S(t)}\left[C_{FP}\left(S(t),K_D(t)\right)+C_{FN}\left(S(t),K_D(t)\right)+C_T\left(S(t),K_D(t)\right)\right].$$

From the other side, when defense efficacy $E(t)$ changes then the attack team chooses such team configuration and behavior that minimize the amount of packet sent $C_P(E(t),K_A(t))$ and the amount of daemons defeated $C_D(E(t),K_A(t))$:

$$\min_{E(t)}\left[C_P\left(E(t),K_A(t)\right)+C_D\left(E(t),K_A(t)\right)\right].$$

The model of computer network is defined as $Network=\{Tp,P,Tr\}$, where Tp – network topology; P – protocols; Tr – traffic.

The proposed integrated *simulation environment (tool)* architecture was implemented using OMNeT++ INET Framework and software modules implemented in C++. It was intended for multi-agent simulation of DDoS attack and defense mechanisms. There are used the following *specification elements to define the network models, attack and defense mechanisms*:

- *Network topology:* quantity and types of hosts, channels between them and their types. The possibility to deploy certain type of application (or agent) depends on host type.
- *Defense team parameters:* quantity of daemons; master's address and port used for interactions; daemon's port used to send attack packets; victim's address and port; time of attack; attack intensity; address spoofing technique.
- *Attack realization parameters:* victim type (application, host or network; one must define the IP-address and port of victim); type of attack (brute force (UDP/ICMP flood, smurf/fraggle, etc.) or semantic (TCP SYN, incorrect packets, hard requests, etc.)); attack rate dynamics (can be constant or variable); adaptation scheme, etc.
- *Defense team parameters:* address of defended host; detector's address and port for interactions; server's reply size and delay time; adaptation scheme (changing of defense mechanisms) depending on attack severity, etc.
- *Defense mechanisms parameters*: deployment location (source, intermediate or defended subnets); the stages the defense method can implement (attack prevention, attack detection, tracing the attack source, attack counteraction); detection technique (misuse and anomaly detection; one chooses one particular method or the set of methods), etc.
- *User team parameters:* quantity of users; server's address and port; time to start; quantity of requests to server, interval between them and their size; interval between connections.
- *Defense team cooperation parameters:* scheme of cooperation.
- *Simulation parameters:* simulation duration; quantity of experiments; initialization of random number generator.

6. EXPERIMENTS

Various experiments were conducted to analyze the possible actions and scenarios of agent teams as well as the applicability and efficacy of the framework and simulation environment developed.

Experiments for investigating the cooperative work of defense teams included simulation of the following mechanisms: DefCOM (Mirkovic et al. 2005), COSSACK (Papadopoulos et al. 2003), "no cooperation", "filter-level", "sampler-level", "full cooperation". The following agent classes are proposed to introduce in compliance with *DefCOM* architecture: "Alert generator" – detects an attack and warns about it other hosts in the DefCOM network; attack is detected if the traffic exceeds some threshold; "Rate limiter" – limits the traffic that is destined to the attack target; "Classifier" – provides selective traffic limiting, tries to classify attack and legitimate packets and to drop the former. *COSSACK* architecture consists of the following agent classes: "snort" prepares the statistics on the transmitted packets for different traffic flows; the flows are grouped by the address prefix. If one of the flows exceeds the given threshold then its signature is transmitted to "watchdog"; "watchdog" receives traffic data from "snort" and applies the filtering rules on the routers. Agent "snort" is based on the agent "sampler", and agent "watchdog" - on the agent "detector". HCF, SIPM and BPS local defense methods were used. Network topology was defined using the power function of frequency distribution of data links and hosts number was 50. For client team the defended server, 10 clients and their requests parameters were defined. Attack team included 10 daemons that realize UDP flood attack on server. Four defense teams are configured in compliance with the mentioned cooperative defense schemas. Conducted experiments showed the effectiveness of various cooperation schemas of defense teams. Attack traffic was sufficiently lowered regarding the start of attack. The best cooperative schema on the basis of output parameters is "full cooperation" (Figure 2, attack traffic is lowest after 450 s). Samplers-agents cooperation played the crucial role in defense.

Different *adaptation schemas of agent teams* were studied. *Adaptation schemes* operate in the following way. Depending on attack state the defense team adapts the parameters of methods and cooperation reducing the defense cost. The most simple and not resource-intensive method is BPS. Defense team starts the defense implementing BPS. When attack is detected the team continues to use the same method, if it allows to neutralize the attack. If it fails, then defense team applies more complicated SIPM method. If it succeeds to stop the attack, then the defense team returns to BPS. If not – it will additionally use HCF. Conducted experiments showed that one can reach the best attack traffic blocking due to defense teams cooperation.

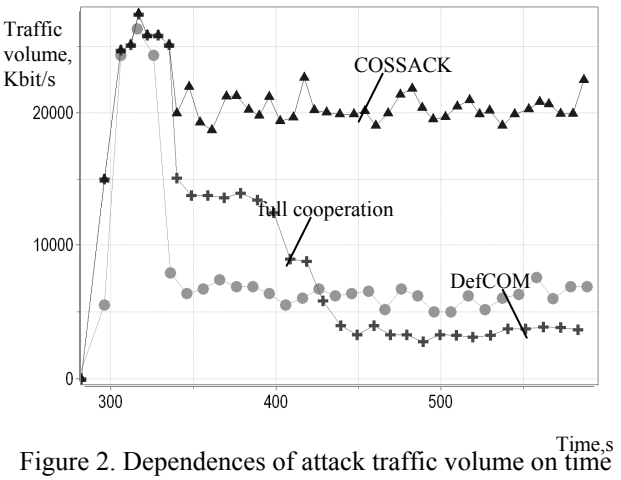

Figure 2. Dependences of attack traffic volume on time for different cooperative schemas

Since sampler cooperation was the determinative in defense it can be used without applying full cooperation during which high teams interaction traffic is observed.

Attack team redistributes the attack intensity between daemons and changes the address spoofing technique to minimize the amount of attack packets and reduce the probability of attack agents' exposure by defense agents. At first the team having many daemons distributes the load equal between them and does not use address spoofing (not to draw suspicion upon themselves from firewall in their subnet). If after the defense team actions some of the daemons will be defeated, the attack team will raise the load to the remaining daemons (to save the given attack intensity) and apply address spoofing technique to avoid detection. If the remaining daemons will not be defeated the team will continue the attack in former mode.

Adaptation schema is tested in various cooperation modes (Figure 3). Cooperative team learning is supposed in sampler-level and full cooperation.

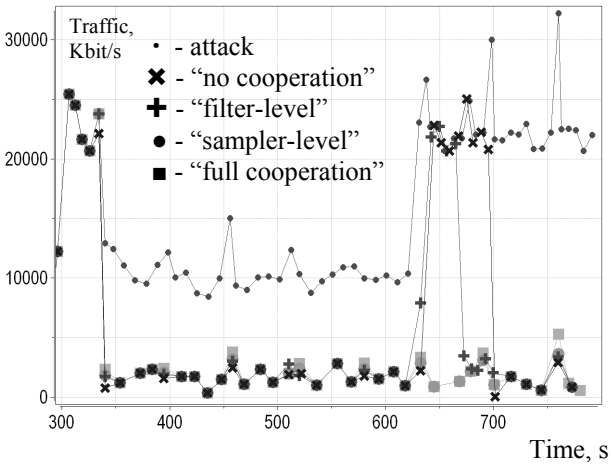

Figure 3. Dependences of attack traffic volume on time using adaptation criteria

7. CONCLUSION

The paper proposed a framework *for multi-agent simulation* of competing teams of intelligent agents. The framework is characterized by attempting to formalize all components needed for supporting agent teamwork and team competing. The *simulation environment* implementing this framework was developed. It is a multi-level software tool which allows to use the various discrete-event simulators and the libraries (ontologies) of different subject domains. The proposed approach was implemented on an example of multi-agent simulation of the Internet attacks and defense mechanisms. The attack and defense teams' competing was implemented. The experiments showed the applicability and efficacy of the framework and simulation environment developed.

Future work is concerned with further formalization of antagonistic and non-antagonistic competing as well as various cooperation schemas of agent teams. Other direction of research is improving the simulation environment. We are supposed to develop and implement improved adaptation and self-learning mechanisms (as the current are prone to manipulation by opponents). The investigation of other applications for the framework suggested and the environment implemented (together with expansion of subject domain library) is very important. The essential part of future research is providing numerous experiments to analyze the framework and simulation environment.

8. ACKNOWLEDGEMENT

This research is being supported by grant of Russian Foundation of Basic Research (Project No. 07-01-00547), program of fundamental research of the Department for Nanotechnologies and Informational Technologies of the Russian Academy of Sciences and partly funded by the EU as part of the RE-TRUST project (contract No. 021186-2). The author thanks Alexander Ulanov and Alexey Alexeev for developing the software and fulfilling a multitude of experiments.

REFERENCES

Chen, S. and Q. Song. 2005. "Perimeter-Based Defense against High Bandwidth DDoS Attacks." *IEEE Transactions on Parallel and Distributed Systems*. 16, 7.

Cohen, P. and H.J. Levesque. 1991. "Teamwork." *Nous*, 35.

FIPA, http://www.fipa.org

Grosz, B. and S. Kraus. 1996. "Collaborative Plans for Complex Group Actions." *Artificial Intelligence*. 86, 2.

Haque, N.; N.R.Jennings; and L. Moreau. 2005. "Resource allocation in communication networks using market-based agents." *International Journal of Knowledge Based Systems*. 18, 4-5.

Jennings, N.R. 1995. "Controlling cooperative problem solving in industrial multi-agent systems using joint intentions." *Artificial Intelligence*. 75, 2.

Kaminka, G.A., and I. Frenkel. 2005. "Flexible teamwork in behavior-based robots." In *Proceedings of AAAI-05.*

Kaminka, G.A.; A. Yakir; D. Erusalimchik; and N. Cohen. 2007. "Towards Collaborative Task and Team Maintenance." In *Proceedings of AAMAS-07*.

Keromytis, A.D.; V. Misra; and D. Rubenstein. 2003. „SOS: An architecture for mitigating DDoS attacks." *Journal on Selected Areas in Communications*. 21.

Kotenko, I., and A. Ulanov. 2006. "Simulation of Internet DDoS Attacks and Defense." *Lecture Notes in Computer Science*. 4176.

Kotenko, I., and A. Ulanov. 2007. "Multi-agent Framework for Simulation of Adaptive Cooperative Defense against Internet Attacks." *Lecture Notes in Artificial Intelligence*. Springer. 4476.

Luke, S.; C. Cioffi-Revilla; L. Panait; and K. Sullivan. 2004. "MASON: A New Multi-Agent Simulation Toolkit." In *Proceedings of the 2004 SwarmFest Workshop*.

Macal, C.M., and M.J. North. 2005. "Tutorial on Agent-based Modeling and Simulation." In *Proceedings of 2005 Winter Simulation Conference.*

Mankins, D.; R. Krishnan; C. Boyd; J. Zao; and M. Frentz. 2001. "Mitigating Distributed Denial of Service Attacks with Dynamic Resource Pricing." *In Proceedings of 17th Annual Computer Security Applications Conference*.

Marietto, M.; N. David; J.S. Sichman; and H. Coelho. 2002. "Requirements Analysis of Agent-Based Simulaton Platforms: State of the Art and New Prospects." *Lecture Notes in Artificial Intelligence*. Springer. 2581.

Mirkovic, J.; M. Robinson; P. Reiher; and G. Oikonomou. 2005. "Distributed Defense Against DDOS Attacks." *Technical Report CIS-TR-2005-02.* Univ. of Delaware.

Network Simulators. *Network Simulators and Related Links*. http://www-nrg.ee.lbl.gov/kfall/netsims.html

Papadopoulos, C.; R. Lindell; I. Mehringer; A. Hussain; and R. Govindan. 2003. "Cossack: Coordinated suppression of simultaneous attacks." In *Proceedings of DISCEX III.*

Robocup Rescue. http://www.robocuprescue.org/

Rosvall, M.; K. Sneppen. 2008. "Modeling self-organization of communication and topology in social networks." *Phys. Rev.,* E 74.

Simulation frameworks. *Simulation software development frameworks*. http://www.topology.org/soft/sim.html

Simtools. *A Collection of Modelling and Simulation Resources*. http://www.idsia.ch/~andrea/simtools.html

Tambe, M. 1997. "Towards flexible teamwork." *Journal of AI Research*. 7.

Tambe, M.; E. Bowring; H. Jung; et al. 2005. "Conflicts in teamwork: Hybrids to the rescue." In *Proceedings of AAMAS-05*.

Wang, H.; and K.G. Shin. 2003. "Transport-aware IP Routers: A Built-in Protection Mechanism to Counter DDoS Attacks." *IEEE Transactions on Parallel and Distributed Systems*, 14, 9.

Xuan, D.; R. Bettati; and W. Zhao. 2002. "A gateway-based defense system for distributed dos attacks in high-speed networks." *IEEE Transactions on Systems, Man, and Cybernetics*.

BIOGRAPHY

IGOR KOTENKO is Professor of computer science and a head of computer security research group in St. Petersburg Institute for Informatics and Automation. His Web-page can be found at `http://comsec.spb.ru/kotenko/`.