Application of Cognitive Techniques to Network Management and Control

Sławomir Kukliński[1,2], Jacek Wytrębowicz[1], Khoa Truong Dinh[1], and Emilia Tantar[3]

[1] Warsaw University of Technology, Warsaw, Poland
slawomir.kuklinski@tele.pw.edu.pl, jwt@ii.pw.edu.pl,
k.truongdinh@stud.elka.pw.edu.pl
[2] Orange Polska, Poland
[3] University of Luxembourg, Luxembourg
emilia.tantar@uni.lu

Abstract. This paper describes the latest communications technologies emphasizing the need of dynamic network control and real-time management operations. It is advocated that many such operations can profit from cognitive learning based techniques that could drive many management or control operations. In that context a short overview of selected networking approaches like 3GPP Self Organizing Networks, Autonomic Network Management and Software-Defined Networking, with some references to existing cognitive approaches is given.

Keywords: machine learning, SON, LTE, SDN, autonomic network management, artificial intelligence.

1 Introduction

In recent years a dynamic growth of communication networks can be observed. This growth concerns especially commercial IP networks and mobile systems, like UMTS (Universal Mobile Telecommunications System) or LTE (Long Term Evolution). Not only the number of network nodes is growing, but also network customer base as well as number of offered services. The services have differentiated traffic requirements and many of them, like video services and cloud processing, require high-speed links and high reliability. The users mobility is another problem that has to be taken into account - the traffic demands change in time and move between different geographic areas. The new situation requires a change of the paradigm of network deployment and maintenance. In the past the deployment of new networks was preceded by careful design and planning in order to optimize long-term investment and at the same time to satisfy present and potentially future (predicted) end-users needs. Unfortunately, the above-mentioned growth of users, services, and traffic related demands make the planning based approach no more efficient or applicable. The solution to this problem is real-time adaptive control of networks, which should enable dynamic allocation of resources and global optimisation of network usage. In response to such demand, concepts like SDN [32], i2rs [5] or GMPLS [31] have arisen. These concepts, implemented already in some networks, are still under evolution; but what is important, they make network control and management programmable. The difference between network control and

A.-A. Tantar et al. (eds.), *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, Advances in Intelligent Systems and Computing 288,
DOI: 10.1007/978-3-319-07494-8_7,

management lies in their purpose: the control is responsible for fast decisions related typically to users behaviour, network management is responsible for network operations and is much slower than network control.

There is however another problem related to the dynamic growth of network nodes, services and end-users. This problem is related to the old-fashioned way in which networks are operated and managed at present. Operators manage networks to configure them properly, handle faults, optimize performance and to cope with security issues. Actually such management is quasi-static, centralized with a key role of human operator. So far only some management operations are automated. The growth of network nodes, services and end-users makes such centralized, human centric management slow, error prone, inefficient and expensive. To cope with this problem, the concept of so-called Autonomic Network Management (ANM) [3] has been born. The concept lies on continuous measurements of the network and execution of suitable decisions according to the network state. Such feedback loop based management has been a subject of investigation of many European projects [9, 13, 42, 47] and is now under standardization in ETSI [18]. The issue of self-management (a synonymous of autonomic network management) is also a hot topic in 3GPP (3rd Generation Partnership Project). In latest generation of mobile systems the concept Self-Organizing Networks (SON) has been proposed [46]. The goal of SON is to automate some of Radio Access Network (RAN) related management operations. This autonomic network management requires fast and adaptive to network state decisions.

The above mentioned technologies like SDN, i2rs, ANM, SON require algorithms, which realize predefined goals of network management or control. It means that a control theory or other real-time optimization techniques are required for their implementation. Fortunately the already mentioned approaches provide the programmability, which is possible not only by the solution manufacturer, but also by network operators, 3rd parties or even by end-users. It should be noted however that the above-mentioned operations are complex ones to realize control and management goals, many (if not all) network devices have to be involved and the complexity of such devices is typically high. Another complication is introduced by the independent optimization of multiple goals. Due to conflicting goals such approach in many cases leads to conflicting control or management decisions. For example minimization of energy consumption is typically against maximization of network performance. Simultaneous optimization of all goals is not being realized by todays management solutions.

There is no doubt that the control theory based or heuristic algorithms can be applied to network control and autonomic management. However, due to complex network behavior, the selection of algorithms or tuning of their parameters is in many cases hard or impossible. A solution of the problem is to use cognitive approaches, i.e., algorithms that have learning abilities. Such approach is very attractive, but also comes with the problem of learning convergence, and the problem of trust to the algorithm behavior. It has to be noted that networks have to operate reliably. Therefore any uncertainty related to algorithm behavior has to be minimized. Despite such constraints, the learning algorithms should be evaluated in the context of network control and management, and such research activity is already observed [9, 13, 15]. In this paper we want to provide an overview of cognitive techniques, which potentially can be applicable to solve network control and management problems. A short overview of cognitive techniques with their specific properties is presented in Section 2. Section 3 describes areas of applications of cognitive techniques, distinguishing radio Self-Organized Networks, autonomous fixed networks, and Software-Defined Networking. The section gives also some proposals of using cognitive techniques, and if applicable, examples of applications where such techniques are already in use. Finally Section 4 concludes the paper.

2 Overview of Cognitive Techniques

The goal of the section is to list some well-known cognitive techniques, which could be applicable to solve networking problems. Each algorithm will be described shortly from the implementers point of view, emphasizing key features of its application.

2.1 Selected Cognitive Techniques Outline

Cognitive techniques include a big set of algorithms, which common denominator is learning. They are so-called neural networks, evolutionary algorithms, machine learning, knowledge based reasoning and many other techniques. The common feature of them is an examination of the environment and taking subsequent actions by analyzing results of previous decisions. Algorithms may learn in different ways. In some of them the learning phase precedes the normal operations in such case we talk about off-line learning. In some cases we have on-line learning. The latter case can be split into continuous learning, in which the algorithm can re-learn; and the case, when during initial phase of operations the algorithm can both operate and learn, and has no re-learning abilities later on. For the off-line learning a set of training data has to be prepared. Such set is typically extracted from real data.

In so-called supervised learning, the training set consists of correct outputs to all input vectors included in it. The error between expected and actual output drives the learning procedure. The learning is iterated until for the whole training set the output error will be reduced to a satisfactory value. An example of a technique, which uses supervised learning, is multi-layer perceptron [39]. A multi-layer perceptron is a feedforward artificial neural network, based on multiple layers, of so-called neurons. Neurons are elements that compute a scalar product of their inputs and their respective weights; finally the scalar product is the input of nonlinear function (i.e., neuron activation function), which typically is sigmoid. The learning is performed using the back propagation (gradient descent) technique. The goal of the learning is to define neurons weights. Learning starts typically with randomized weights and is known to be slow. Normal operations of multi-layer perceptrons are generally fast, i.e. decisions are taken in one pass only. It is expected that after learning the perceptron will properly respond to unknown input, i.e. to the input that is not included in the training set. For proper behaviour of the network it is important to provide a representative set of input-output pairs in the training set. There are no strict rules for the design of multi-layer perceptron, especially in regard to the number of so-called hidden layers and their size. The key parameter of the learning, so-called learning rate may impact learning convergence and results. The re-learning is not possible in this method. Typically multi-layer perceptron is used as a classifier.

Another technique that uses supervised learning is called Support Vector Machines [14] used for linear or non-linear classification and regression analysis.

In opposite to supervised learning also unsupervised learning exists. In such approach the training set should contain only input data, no input-output pairs as in supervised learning are used. This type of learning is used to find the hidden structure of the input data set, and is closely related to statistical density estimation. The best-known approach of such learning is Self-Organizing Maps concept [28]. This technique is fast, but the analysed data set cannot be dynamically changed, it has to be prepared in advance. In case of changes in the input set, the algorithm has to restart. Unsupervised learning is typically used for data clustering. In recurrent neural networks, like Hopfield networks [23], the output of the neuron, which is similar to that used in multilayer perceptron, but typically has a binary activation function, is fed back to its inputs with predefined weights as well as the input vector. During iterations (synchronous or asynchronous ones) this type of network goes towards equilibrium (if certain conditions are satisfied). The Hopfield network, which weights are predefined by the Hebb rule, is typically used as Content Addressable Memory (CAM). CAM is able to restore incomplete input patterns. For the Hopfield network an energy scalar can be assigned, and during iterations the energy cant increase, and typically should go down. This property is used for application of the Hopfield network for solving optimization problems, like the travelling salesman problem. The main problem of the application of this network is the convergence time and potential lack of stability.

Reinforcement Learning (RL) is a type of machine learning technique that exploits the learning agents experience in order to learn the optimal behaviour in an environment [44]. Through its interaction with the environment, the agent tries to learn actions for particular states of the system, so that the long-term rewards are maximized. RL learns its actions on basis of the observation of the environment; however each action impacts the environment. RL performs an online search to find an optimal decision policy in multi-stage decision problems. The learning problems are usually modelled as Markov Decision Processes [40] and solved using Dynamic Programing techniques. In opposite to other learning approaches, RL evaluates the taken actions rather than corrections to actions. In combination with Fuzzy Sets Theory [49], the RL-FIS (Fuzzy Inference System) is widely applied in real world to different control processes, especially its model-free based learning variant (Q-learning). RL approach is typically used for continuous real-time learning. Multi-agent RL approaches can be used to simultaneously solve multiple problems.

In Case-based Reasoning (CBR) [2] possible actions are based on the past experiences in similar circumstances. For each case (input) the status of the corresponding environment as well as taken actions are memorized. During the analysis of a new case the prior cases are looked for similarity. If such similarity is found a corresponding, already memorized, action is taken. If there is no good match, a new action is created for such a new case. The typical CBR process lies on looking for the similarity with other cases and if found, taking appropriate action(s). If no similarity is found, that case is adapted and retained. For proper operation (correct generalization) the prior cases should be statistically relevant.

Particle algorithms [16] are an extensively used computational tool not only in engineering, but also in machine learning, statistics and physics. Nevertheless, various denominations have been developed simultaneously (according to the context where they were employed), ranging from: particle filters, bootstrap or genetic filters, population Monte Carlo methods, sequential Monte Carlo models, genetic search models, branching and multi-level splitting or particle rare event simulations. Evolutionary Algorithms (EA) [7] represent a heuristic approach inspired by natural evolution, having as main applicative area optimization. They include Genetic Programming [29,48], Genetic Algorithms [19] and Evolutionary Strategy [8]. According to [38], in the EA area the algorithmic development can be divided into two main paths: direct and indirect search methods. Direct methods rely on direct sampling of the objective function and the search is conducted by applying a set of perturbations on the initially generated set of samples (often randomly generated). Examples in this direction include: Hill-Climbing, Nelder-Mead, Solis and Wets, Tabu Search, Variable Neighborhood Search, Simulated Annealing, Iterated Local Search. Indirect methods suppose, in a continuous optimization context, the use of second order derivatives.

Besides the alrady mentioned single solution based approaches, set oriented methods emerged, their foundation being due to Holland [22]. Several variants exist, including among others, Scatter Search, Differential Evolution, Artificial Immune Systems or Swarm Intelligence, e.g. Bee Colony and Ant Colony Optimization. In addition to stand-alone algorithms, various hybrid versions were developed thus making place for meta-heuristics and later, hyper-heuristics. Another trend supposes the use of additional tools or information. Examples are the Covariance Matrix Adaptation [20], relying on information obtained from the landscape of feasible solutions; or Estimation of Distribution Algorithms [30] that construct a probabilistic model in order to capture the traits of solutions that fit a set of given criteria, e.g. fitness below a specified threshold. Multiple criteria can also be treated simultaneously, this being covered by the multi-objective optimization area. An extensive bibliography in the area is available at [12].

These techniques are applied mainly offline, nevertheless a specific branch of dynamic optimization handles contexts involving the use of dynamically evolving environments.

There are also other algorithmic techniques, which can be nicely combined with cognitive ones. Such techniques include principal component analysis [25], Fuzzy Sets [49], Rough sets [36], clustering techniques like k-means clustering [21], k-nearest neighbour clustering [26], Bayesian networks [24], Hidden Markov Models [37], Markov Decision Processes, or discovery of frequent patterns [10]. In general, using some pre-processing of input data may help in having a more efficient implementation of cognitive algorithms, reducing their learning time and improving accuracy.

2.2 Drawbacks of Cognitive Approaches

The main value of cognitive approaches is that network operators, who dont know how to solve a given problem or the problem is too complex to be solved using analytical tools, can use these approaches to find a solution. Although it should be noted that optimization and learning techniques will make use of a simplified model of the treated problem, thus inherently loosing some accuracy through the description employed.

However such approaches come also with some problems related to their proper application, and selection of many learning algorithm parameters. This includes for example so-called learning rate, exploitation/exploration parameter, neural network topology or initial pre-processing of input data. It has to be noted, that an expert knowledge can be used for input data pre-processing as well as for evaluation of results.

Another problem is related to the algorithm convergence time, especially when the algorithm is applied in real-time, although techniques with proven performance guarantees in theory and practice emerge, see [16]. In some recursive approaches there is potential danger of unstable (chaotic) behaviour of the algorithm. However the most important problem with many cognitive algorithms lies on the lack of trust that they behave properly if new situation will happen, and that they provide high overall quality of results in presence of uncertainty.

3 Areas of Applications of Cognitive Techniques in Networks

The cognitive techniques have huge potential; however until now their usage for solving communication networks problems (control and management) is rather low. In this section we outline potential application areas of cognitive techniques in communication networks. We also propose some candidate techniques for solving selected problems. If a cognitive approach has been already proposed, by a research paper or an implementation, this is also mentioned in this section.

3.1 Application of Cognitive Techniques for SON Enabled Mobile Networks

At present very dynamic development of mobile networks is observed. It can be also noted that a new generation of mobile systems is typically introduced in several years after the deployment of the previous one. The users of these networks require more and more bandwidth.

The most problematic part of each mobile system is so called Radio Access Network (RAN). A widely accepted approach to RAN lies on cellular structure of the radio network, and one hop communication between user terminals and one of the radio base stations. In Europe, average operators RAN is composed of thousands of radio base stations to provide nationwide coverage and appropriate capacity to handle users traffic demands. It is worth to note that in the latest generations of mobile systems, the typical cell size is smaller than in previous ones in order to handle more traffic; therefore the number of cell operated by the operators grows. The cellular structure of RAN imposes also the issue of users mobility handling, i.e. seamless switching users radio links between neighbouring cells without interruption of active communication sessions. The mechanism is known as handover.

At present every operator deploying new network has to configure properly each base station, and during its operation has to optimize RAN performance and handle possible faults. The configuration process is so far manual and static, performance optimization is very limited, and fault handling has limited automation level. It is worth to emphasize that manual management of such network is costly, error prone, slow and not scalable. In order to cope with the mentioned problems and scalability imposed by growing number of base stations, the 3GPP organization, which standardizes GSM, UMTS and LTE, has proposed a solution named SON (Self-Organizing Network) [46], in which certain management processes of RAN are fully automated. SON functions have been originally defined in 3GPP Release 8, which describes the LTE network, but it is also expected that they will evolve and also retrofit the previous generation, i.e., UMTS. At present 3GPP has only provided a list of SON functions, and so far the way of their implementation is manufacturer dependent. There is no doubt, that these functions require adaptive control, and that the cognitive techniques can be nicely used for SON.

SON Functions and Cognitive Techniques. All SON functions can be divided into three following groups: self-configuration functions, self-optimization functions and self-healing functions. The main goal of self-configuration functions is to obtain the Plug-and-Play base station deployment. They are: Configuration of Initial Radio Transmission Parameters, Neighbor Relation Management, Automatic Connectivity Management, Self-testing, and Automatic Inventory.

Automatic Configuration of Initial Radio Transmission Parameters lies on individual self-tuning of base station parameters, to provide proper coverage and minimize interferences among base stations. Cognitive approach to this problem has been already used. The RL combined with a multi-agent implementation has been proposed to address the problem of Inter-Cell Interference Coordination in the downlink channel [17]. In the paper, each base station is considered as an agent, which dynamically changes its transmit power to control the interference with the neighbouring cells. In the described approach FIS rules were used to handle the continuous input and output space. Moreover, an improved effective initialization procedure was provided to overcome slow convergence the drawback of Q-learning. The RL approach based on Q-Learning framework has been also proposed to handle channel sharing between small cells deployed in a macro cell [41].

Automatic Neighbour Relation Management concerns the updates of neighbour cell relationships to facilitate easy handovers between base stations. It is necessary to have up to date neighbour list to avoid dropped calls, failed handovers and QoS degradation. The manual update of neighbour relationships become not scalable and error prone, therefore it is automated in SON. The operation is very important for seamless RAN enhancement. However, it does not require cognitive approaches.

Automatic Connectivity Management lies on connectivity automation of initially deployed base station. The cognitive approach is not required in this case.

Self-testing is a process used for verification of proper functioning of base station modules. The cognitive techniques can be applied for detection of anomalies. However, the base stations internal architecture is manufacturer dependent, therefore such algorithms can only be deployed by the equipment manufacturer.

Automatic Inventory collects information about base station internals (hardware boards and software components). No cognitive techniques are applicable here.

The self-optimization SON functions are: Mobility Robustness Optimisation, Mobility Load Balancing and Traffic Steering, Energy Saving, Coverage and Capacity Optimisation, and Random Access Channel Optimisation.

- Handover (HO) is one of the key procedures for ensuring that users can move freely through the network while staying connected and being offered appropriate service quality. It is important that HO procedure happens as timely and seamlessly as possible. For quality of the handover process parameters like Radio Link Failure, Handover Failure and Handover Ping-Pong are used. The aim of **Mobility Robustness Optimisation** is to tune simultaneously handover parameters, like time-to-trigger or handover hysteresis, on the basis of previously mentioned handover performance indicators, in order to achieve optimal HO decision. Cognitive techniques are applicable to solve the problem.
- Due to the users mobility, the load of different BSs is rarely uniform. As a result of limited capacity of the network, the potential congestion may occur, which might degrade users QoS. To cope with this problem, the **Mobility Load Balancing and Traffic Steering** mechanism has been proposed. This mechanism uses information about relative load of neighbouring cells in order to slowdown or speedup HO. In [33] a distributed Q-Learning approach is used to dynamically select the HO parameters according to BSs load.
- The main purpose of **Energy Saving** lies on seamlessly turning off and on some radio nodes (BSs), sectors carriers or nodes internal blocks, to reduce the power consumption when the network load is very low (typically at night). The decision about switching an element off/on can be based on cognitive techniques. The RL algorithm, that is a model free, can be applied as in [13]. In the proposed approach each BS collects monitoring data and sends periodic report to its network cognitive manager. The manager takes the decision about switching off or on a particular BS, on the base of configuration, network topology, policy parameters and previous decision history. Cognitive techniques are appropriate one to handle this problem. However, improper behaviour of an algorithm could lead to subsequent turning off the recently turned on BS, and to QoS degradation of active data streams.
- The BS configuration parameters initially set, next can be dynamically modified by the **Network Coverage and Capacity Optimisation** function. The decision about changing BS parameters (like transmission power, antenna tilt, etc.) can be triggered by the quality reports obtained from users. The reports include such parameters like signal-to-noise ratio, timing advance or radio link failures. The improvement procedure can be based on cognitive techniques, which can use reinforcement learning. The classification of quality perceived by users can apply unsupervised learning, to reduce complexity (the input space dimension).
- In LTE mobile networks, Random Access Channel (RACH) is an uplink channel, used for initial access of a user. This is (according to the name) a random access procedure with possible conflicts. The performance of RACH procedure is dependent on network configuration and load. Measurements of RACH performance (overload), cell load and number of users, can be used in SON for **RACH Optimization** function, by the use of a cognitive technique.

The self-healing SON functions are: Cell Degradation Monitoring and Management, and Cell Outage Compensation.

Cell Degradation Monitoring and Management lies on the detection of a fault, by analysing some monitoring data available at each BS, collected from neighbouring

BSs or from users. Once a failure is detected an alarm is triggered, and if possible a healing action is taken. There are two important problems related with this function. The first one is how to discover that the measured state is degraded. The second is how to heal the network. Both problems can be nicely solved using cognitive techniques. The first one is in fact anomaly detection problem, which can be solved by unsupervised learning, whereas the second one is much complex because the healing action may involve multiple devices and mechanisms.

Cell Outage Compensation lies on the compensation of outage of a cell. The healing process has to be fast and the compensation must come from the neighbouring cells, what has to be achieved by increasing the neighbouring cells power, adjusting the antenna tilt to cover the area of the failed one. This procedure is similar to procedures of Coverage and Capacity Optimisation or Energy Saving function.

Coordination of SON Functions. The described SON functions, until recently, have been treated separately. However, in real networks they have to operate simultaneously in order to optimise the overall RAN performance. With the increasing number of SON functions, the probability of conflicts and dependences between them increases.

Such conflicts can occur if two individual SON functions aiming at different goals (e.g., Cell Outage Management and Energy Saving) try to modify the same parameter (e.g., the BS transmit power). There is also a conflict, if the modification of a triggering parameter by one SON function has negative impact on other SON functions. These conflicts may cause that the entire system will operate far from the optimal state, and the network performance and users satisfaction will be degraded. An overview of conflicts between different SON functions has been provided in the SOCRATES project [42].

As more and more SON functions are being standardized, the coordination between them is becoming more important in order to prevent and resolve unexpected and undesirable network behaviour. The coordination has to impact all individual SON functions, and at the same time to enable the operator supervising the overall SON system, and to apply operators preferences (policies). With the increasing number of SON functions, the input data space and output data space become large and multi-dimensional. The result is that mapping from input and output parameters to the optimisation objectives is very difficult to define. Moreover, the information about the input may be incomplete or not up to date (due to transmission delays, faults and dynamic changes of network state), which can cause negative impact to the decision-making process. Actually there is an on-going work on the general solution for the coordination problem, but there are no yet presented results. There are some approaches for simultaneous optimization of several SON functions. An integrated approach for two SON functions implementation, namely handover optimisation and load balancing combined with admission control is presented in [45]. In the paper the Fuzzy Q-Learning Control (FQLC) algorithm was combined with a heuristic algorithm. Both algorithms operate in different time scales.

3.2 Application of Cognitive Techniques for Management of Fixed Networks

As the number of nodes and the size of the network increase, and functionality of the devices grows, the centralized management becomes inappropriate. Hence distribution of management functions as a solution is proposed, as well as adaptation of management decisions to network status. A new approach is based on a continuous feedback from the network after making management decisions. The concept has been originally proposed by IBM for autonomic management of IT resources and described as the Monitor-Analyse-Plan-Execute (MAPE) model [1].

Using MAPE, several architectures have been proposed for network management and labelled as Autonomic Network Management (ANM). ANM includes fault handling, performance optimization (which includes energy saving approaches, configuration or dynamic reconfiguration of devices), and management of security. In general the ANM operations should concern all OSI ISO [Open System Interconnection defined by International Organization for Standardization] layers. During last six years a number of research projects has been launched, with the aim to find efficient solutions for automatic or even autonomic management of future networks. Below, we mention those, which used cognitive approaches in different contexts of network management.

The FOCALE autonomic architecture [43] defines maintenance and reconfiguration loops and contains among other reasoning and learning blocks. Data retrieved by sensors are normalized and analysed. If the managed resource follows expected states (defined by its model) the system continues using the maintenance loop. If a problem is detected, then the reconfiguration loop is used. The Autonomic Internet (AutoI) project [6] has aimed to design and develop a self-managing virtual resource overlay for heterogeneous networks. The overlay supports service mobility, QoS, and reliability. It used ontology-based information and data models, providing fast and guaranteed service delivery.

In the framework of the EFIPSANS (European 7th Framework Project), the Generic Autonomic Network Architecture (GANA) has been developed. Next ETSI has adapted and extended GANA as a Reference Model for autonomic network engineering, cognitive networking and self-management [11]. The model defines control loops for: protocol, function, node, and network levels. At every level a rich set of decision elements operates. Every decision element acts in a control loop. Algorithms are no subjects of GANA standardization. The BIONETS project [9] has aimed to find out paradigms taken from biology, physics and social science, which can be applicable for autonomic networks and services. In the project not only techniques for network management were analysed but also those, which are applicable for network control, e.g. routing, forwarding in Delay Tolerant Networks, and congestion control. In that context some cognitive techniques have been analysed [4].

The UniverSelf project [47] has aimed to design of a Unified Management Framework (UMF) for wireless and wireline networks and services. UMF has been intended to be equipped with all functions required to achieve complete self-management of networks. There are some UMF elements with learning abilities and knowledge based reasoning. The COMMUNE projects [13] goal is to cope with network management under uncertainty using cognitive techniques. The project consortium published many use cases of application of cognitive techniques to FTTH (Fiber to the Home), LTE, M2M (Machine to Machine) networks, as well as to video streaming.

3.3 Application of Cognitive Techniques to SDN

As it was outlined, in some ANM approaches the cognitive techniques are already applied. Unfortunately none of them has a commercial deployment yet. Moreover, there is no common architecture framework, for easy implementation of different cognitive techniques.

Software-Defined Networking has recently become the most important network research area, and is seen as the largest revolution in IP networks. At the present most of SDN solutions are based on the OpenFlow protocol [32], defined by Open Networking Foundation [34]. The fundamental idea of SDN networks lies on the separation of the control plane, which decides where to forward the incoming data packets/flows, and forwarding plane, which delivers the traffic to the destination. The control plane is logically centralized, and in small networks only one controller takes decisions related to all control operations. The data plane in the ONF model is composed of relatively simple switches, which main role is to forward flows according to controller decision. An important property of this approach is high-level programmability of the controller, which functions are not predefined, and are relatively simple. An OpenFlow switch can compare and alter IP packet headers, according to the controller instructions. The programmability of SDN makes the network to be application-aware, and increases the customizability and flexibility of deployment.

Following the ONF specifications [35], the applications plane proves to be a gold mine in integrating cognition, as several types of applicaitons will be needed, handling from security concerns, to performance and sustainable abilities; not to mention business specific applications. Providing efficient answers in such a dynamic environment requires the ability of learning from the previous observed network behavior and provide solutions accordingly; while having a clear separation between the SDN infrastructure management and the actual network management.

The above-mentioned programmability requires many algorithms to support network operations. The SDN controller performs many basic and also advanced tasks. The tasks include network topology discovery, network monitoring, creating of network connectivity matrix, and admission control of the user traffic. The SDN controller should perform centralized traffic engineering operations, which provide load balancing taking into account network state, achieving that way optimal usage of network resources and maximizing users satisfaction.

The controller should also perform self-management of the network, handle network topology changes and failures, and last but not least should provide basic detection of DoS security attacks. All the above mentioned operations should fulfil operator policy requirements.

There are already some approaches, which use autonomic or cognitive techniques to solve SDN problems. For example Kim, in his doctoral thesis [27], proposed an autonomic network management architecture called CogMan for SDN. The proposed approach provides reactive, deliberative, and reflective loops, where the reflective loop is intended to use learning algorithms for reasoning and for influencing the planning and decisions performed in the deliberative loop.

Centralized Traffic Engineering in SDN. The traffic engineering in SDN may benefit from the global network view. It should collect data about the network state from switches, signaled users traffic demands, and react to operator policies. With such information the Central Traffic Engineering (CTE) engine determines path assignments for individual traffic flows or group of flows, taking into account the forwarding class, the virtual price of the path and load distribution in the network. Flow level operations enable efficient dissemination of the traffic in the whole network. CTE has to provide fairness for users of the same classes of network services. In case of network topology change or failure, CTE should react quickly and recalculate relevant paths. In order to avoid the overall congestion of the network, CTE can use the admission control mechanism. For efficient usage of available network resources, CTE should apply dynamic load balancing; which should use information about network topology, as well as load of all network nodes and links. The forwarding rules should concern traffic with or without QoS guarantees, using traffic descriptors signaled by users or deduced from network observations.

CTE is a complex problem that has to be solved in real-time (the response time in range of 100 ms) for many thousands of flows. CTE algorithms should be adaptive, i.e., they should monitor some network parameters and take some actions according to their values (a feedback based approach). In relatively simple cases such algorithms can be defined a priori. But in many complex cases the proper design of algorithms requires a lot of efforts and the knowledge about the underlying mechanisms and their impact on the network behaviour. The cognitive approach is seen as a potential solution aiming to solve this problem. In such an approach, appropriate algorithms with learning abilities have to be used to achieve the predefined goals. Due to learning abilities there is no need to analyse all possible situations in fact the cognitive algorithm will adapt according to the specific network condition. The complexity of CTE suggests the usage of multiple cognitive algorithms, which have to be coordinated (see 3.1.2). So far there is none cognitive approach to traffic engineering in SDN.

DoS Attack Detection. SDN network provides the separation between the control plane and the data plane. As a result, the network offers more programmability, flexibility and customizability. The controller, with the central view, offers deeper level of granularity to packet analysis, better global network monitoring. Therefore, SDN paradigm can enhance the security level of the network. In an OpenFlow network, each switch has forwarding tables, which contain the rules for incoming packets/flows. However, in the case of reactive flow handling, the switch does not know the rule for an incoming flow, therefore a request will be directed to the controller to obtain flow rules table entry. If a network node is compromised, and constantly sends requests, then the controller will be a congestion point in the network. In case the controller is compromised, the attacker can generate fake flow controls to the network nodes, containing useless forwarding rules and overflow switches forwarding tables, or directs the flows to wrong destinations. Such situations are called DoS attack, where either controller or switches are blocked.

There is a significant need for DoS attack detection and protection in SDN. A cognitive approach that analyses traffic statistical information and different types of alarms, could detect any unusual events (i.e. abnormal traffic patterns appeared in the network). Moreover, by learning from experience, when there is a high volume of packet/flow sending in the network, the cognitive approach can detect more efficiently what is abnormal.

4 Conclusions

The flexible management and control of communication networks is an extremely important topic. Service providers, network operators, and device manufacturers look for new solutions, which enable to overcome limitations of today technologies in terms of flexibility and easiness of deployment of new services, with reduced cost of operations. Some of the mentioned in the paper technologies like SON, ANM or SDN come with such promise, however they are still not mature yet. A key role in the concepts will perform algorithms, which will drive the whole network behaviour. The described in the paper technologies are programmable not only by the manufacturer but also by operators and even by end-users. It creates a new market for software applications. In our opinion the cognitive techniques can be nicely applicable to solve many control and network management problems. The paper is an attempt to indicate in which networking areas such techniques can be applied. However the existing number of possible mechanisms, which can profit from these techniques, is so huge that selection of appropriate cognitive algorithm starts to be challenging. As it has been noticed there are already some attempts to use cognitive techniques for network control and management.

Acknowledgment. This publication is based in parts on work performed in the framework of the CoSDN project, INTER/POLLUX/12/4434480, funded by NCBiR, Poland and the Fonds National de la Recherche, Luxembourg.

References

[1] An architectural blueprint for autonomic computing. Tech. rep., IBM (June 2005)
[2] Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Commun. 7(1), 39–59 (1994)
[3] Agoulmine, N.: Autonomic Network Management Principles: From Concepts to Applications. Elsevier Science (2010)
[4] Altman, E., Dini, P., Miorandi, D.: Paradigms for biologically- inspired autonomic networks and services the bionets project ebook. eBook (BIONETS project EU-IST-FET-SAC-FP6-027748 project deliverable D0.2.3 (2010), www.bionets.eu
[5] Atlas, A., Nadeau, T., Ward, D.: Interface to the Routing System Framework. draft-ward-irs-framework-00 (July 2012), http://tools.ietf.org/html/draft-ward-irs-framework-00
[6] Autonomic Internet Project, http://ist-autoi.eu/
[7] Bäck, T., Fogel, D., Michalewicz, Z. (eds.): Handbook of Evolutionary Computation. Institute of Physics Publishing Ltd., Bristol and Oxford University Press, New York (1997)
[8] Beyer, H.G., Schwefel, H.P.: Evolution strategies - a comprehensive introduction. Natural Computing 1(1), 3–52 (2002)
[9] BIONETS, http://www.bionets.eu/

[10] Cabaj, K.: Frequent events and epochs in data stream. In: Kryszkiewicz, M., Peters, J.F., Rybiński, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 475–484. Springer, Heidelberg (2007)
[11] Chaparadza, R.: Requirements for a generic autonomic network architecture (gana), suitable for standardizable autonomic behavior specifications for diverse networking environments. Annual Review of Communications 61 (2008)
[12] Coello Coello, C.A.: List of references on evolutionary multiobjective optimization (1999), http://www.lania.mx/~ccoello/EMOO/EMOObib.html
[13] COMMUNE Celtic Project, http://projects.celtic-initiative.org/commune
[14] Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. 20(3), 273–297 (1995)
[15] COSDN, http://secan-lab.uni.lu/index.php/projects/cosdn
[16] Del Moral, P., Tantar, A.-A., Tantar, E.: On the foundations and the applications of evolutionary computing. In: Tantar, E., Tantar, A.-A., Bouvry, P., Del Moral, P., Legrand, P., Coello Coello, C.A., Schütze, O. (eds.) EVOLVE- A bridge between Probability, Set Oriented Numerics and Evolutionary Computation. SCI, vol. 447, pp. 3–89. Springer, Heidelberg (2013)
[17] Dirani, M., Altman, Z.: A cooperative reinforcement learning approach for inter-cell interference coordination in ofdma cellular networks. In: WiOpt, pp. 170–176. IEEE (2010)
[18] European Telecommunications Standard Institute (ETSI), http://www.etsi.org
[19] Goldberg, D., Korb, B., Deb, K.: Messy Genetic Algorithms: Motivation, Analysis, and First Results. Complex Systems 3(5), 493–530 (1989)
[20] Hansen, N., Ostermeier, A.: Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In: International Conference on Evolutionary Computation, pp. 312–317 (1996)
[21] Hartigan, J., Wong, M.: Algorithm AS 136: A K-means clustering algorithm. In: Applied Statistics, pp. 100–108 (1979)
[22] Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. MIT Press, Cambridge (1992)
[23] Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences of the United States of America 79(8), 2554–2558 (1982)
[24] Jensen, F.V.: Introduction to Bayesian Networks, 1st edn. Springer-Verlag New York, Inc., Secaucus (1996)
[25] Jolliffe, I.: Principal Component Analysis, 2nd edn. Springer (October 2002)
[26] Keller, J.M., Gray, M.R., Givens, J.A.: A fuzzy k-nearest neighbor algorithm. IEEE Transactions on Systems, Man, and Cybernetics 15(4), 580–585 (1985)
[27] Kim, S.: Cognitive Model-Based Autonomic Fault Management in SDN. Ph.D. thesis, Pohang University of Science and Technology (2013)
[28] Kohonen, T.: Self-organized Formation of Topologically Correct Feature Maps. In: Neurocomputing: Foundations of Research, pp. 509–521. MIT Press, Cambridge (1988)
[29] Langdon, W.B., Poli, R.: Foundations of genetic programming. Springer (2002)
[30] Larrañaga, P., Lozano, J.A. (eds.): Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer, Boston (2002)
[31] Mannie, E.: Generalized Multi-Protocol Label Switching (GMPLS) Architecture. RFC 3945 (Proposed Standard) (October 2004), http://www.ietf.org/rfc/rfc3945.txt
[32] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: Openflow: Enabling innovation in campus networks. SIGCOMM Comput. Commun. Rev. 38(2), 69–74 (2008)

[33] Mwanje, S., Mitschele-Thiel, A.: A q-learning strategy for lte mobility load balancing. In: Proceedings of the 24th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2013), London, UK (September 2013)
[34] Open Networking Foundation, http://opennetworking.org
[35] Open Networking Foundation: SDN architecture overview. ONF White paper (December 2013)
[36] Pawlak, Z.: Rough set theory and its applications to data analysis. Cybernetics and Systems 29(7), 661–688 (1998)
[37] Rabiner, L., Juang, B.: An introduction to hidden Markov models. IEEE ASSP Magazine 3(1), 4–16 (2003)
[38] Rao, S., Shantha, C.: Numerical Methods: With Programs in BASIC, FORTRAN, Pascal and C++. Universities Press, India (2004)
[39] Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning Internal Representations by Error Propagation. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, pp. 318–362. MIT Press, Cambridge (1986)
[40] Rust, J.P.: Structural estimation of markov decision processes. In: Engle, R.F., McFadden, D. (eds.) Handbook of Econometrics, 1st edn., vol. 4, ch. 51, pp. 3081–3143. Elsevier (1986)
[41] Simsek, M., Czylwik, A., Galindo-Serrano, A., Giupponi, L.: Improved decentralized q-learning algorithm for interference reduction in lte-femtocells. In: Wireless Advanced (WiAd), pp. 138–143 (June 2011)
[42] SOCRATES, http://www.fp7-socrates.eu
[43] Strassner, J., Agoulmine, N., Lehtihet, E.: FOCALE: A Novel Autonomic Networking Architecture (2006)
[44] Sutton, R., Barto, A.: Reinforcement learning: An introduction. MIT Press, Cambridge (1998)
[45] Truong, K., Kuklinski, S.: Joint implementation of several lte-son functions. In: GLOBECOM Workshops, Atlanta, USA (December 2013)
[46] Universal Mobile Telecommunications System (UMTS); LTE; Telecommunication management; Self-Organizing Networks (SON); Concepts and requirements. 3GPP TS 32.500 (version 8.0.0, Release 8) (December 2008)
[47] UniverSelf, http://www.univerself-project.eu
[48] Walker, M.: Introduction to Genetic Programming (October 2001), http://www.cs.montana.edu/~bwall/cs580/introduction_to_gp.pdf
[49] Zimmermann, H.J.: Fuzzy Set Theory and its Applications. Springer (2001)