DYNAMIC CONTROL OF NETWORK PROTOCOLS
A New Vision for Future Self-organising Networks

Sven Tomforde, Emre Cakar and Jörg Hähner
Institute of Systems Engineering, Leibniz Universität Hannover, Appelstr. 4, 30167 Hannover, Germany

Keywords: Organic Computing, Data communication network, Protocol, Parameter, Optimisation.

Abstract: In recent years communication protocols have shown an increasing complexity, in particular in terms of the number of variable parameters. Data communication networks like, e. g. , the Internet reach the limits of their extensibility which leads to initiatives coping with the future of the Internet and data communication in general. A first step towards creating a sustainable solution without exchanging the whole system is to make the static character of network protocols more flexible. An adaptive behaviour of nodes within a network and an autonomous, self-organising concept for their control strategies leads to a possible increase in performance accompanied by an increase of extensibility. This paper presents a new vision of how to establish these new control strategies mostly independent of the particular protocol by using the concepts of Organic and Autonomic Computing. We introduce an adaptive and automated network control system for the dynamic and self-organised control of protocol parameters. This system consists of two sub-systems: an on-line adaptation mechanism and an off-line learning component. The current status is introduced in combination with the definition of further challenges and fields of research.

1 INTRODUCTION

Recent years were characterised by a dramatical growth of communication need and increase of traffic over data communication networks. In combination with the ascending number of protocols and their varying configuration possibilities (configuration space) the complexity of the control task at each node in the network is growing. Based on this observation, more and more often the question arises whether the current structure of the network (in particular the Internet) will be able to cope with the increasing demand (cf. e. g. (Handley, 2006)). This leads to a new vision for the future of the Internet ((Siekkinen et al., 2007)).

The number of researchers formulating the need to exchange the complete set of techniques (e. g. protocols, structure, etc.) is increasing steadily. One major problem focused here is that the existing protocols are designed as static solutions. Although the situation at particular nodes within the communication networks (in terms of e. g. resource usage, available bandwidth, currently known neighbours, etc.) changes over time, the configuration is typically not adapted to the current requirements. A possibility to solve this problem by keeping downward compatibility (this means cooperation of static and dynamic solutions) is presented within this paper.

Based on the approaches of Organic Computing (OC - cf. (Schmeck, 2005)) and Autonomic Computing (AC - cf. (Kephart and Chess, 2003)) an adaptive network control system is introduced which aims at coping with the large configuration space. The system is locally organised, adaptive, and has learning abilities guaranteeing the best possible performance for each node.

This paper presents an adaptive and automated system for the dynamic and self-organised control of network protocol parameters (e. g. values for timeouts, maximum number of re-transmissions, number of open connections, etc.). Section 2 focuses on a short overview of already introduced approaches to optimise network settings and concludes with the statement that no system with the required properties exists yet. In Section 3 the architecture and general approach for the proposed system are described combined with details on the technical realisation. Afterwards, Section 4 defines the challenges for the further development and the research to be done until a real-time operation can be applied. Finally, Section 5 concludes the vision for dynamic network control.

Tomforde S., Cakar E. and Hähner J. (2009).
DYNAMIC CONTROL OF NETWORK PROTOCOLS - A New Vision for Future Self-organising Networks.
In *Proceedings of the 6th International Conference on Informatics in Control, Automation and Robotics - Intelligent Control Systems and Optimization*, pages 285-290
DOI: 10.5220/0002243302850290

2 STATE OF THE ART

The dynamic selection of network protocol parameter settings depends on the situation-based generation of these settings. Therefore, the task can be divided into two different subtasks: the off-line optimisation of parameter settings for a given (observed) situation and the on-line adaptation of the network controller settings with a suitable parameter set.

The optimisation of parameter settings deals with the problem to determine a set of parameters for a given protocol that is as close to the optimum as possible. The task is characterised by the required amount of time and the quality of the solution to be found. In this context *off-line* means evaluating new possible settings using simulation and thus without interfering with the live-system.

There are several examples where authors optimised the settings of their particular protocols, but their intention has been to optimise a specific protocol and not to create a generic system. Considering the techniques used in our system, the approach of Montana and Redi is connected as they also use an Evolutionary Algorithm (EA) to optimise a full custom communication protocol for military MANETs (Montana and Redi, 2005). A similar optimisation of a protocol (for underwater communications) using an EA is described by Sözer et al. (Sözer et al., 2000). Turgut et al. discuss the usage of a Genetic Algorithm to optimise their MANET-based clustering protocol in (Turgut et al., 2002). They all compare their achieved results to a manual optimisation. In contrast to the network control system presented in this paper the approaches are specific to the particular protocols, but do not aim at providing a generic system which is adaptable to different protocol types.

Due to the time-intensive process of generating optimal parameter sets an on-line usage in live-systems is not applicable. Hence, such a solution has to somehow combine the strengths of optimisation techniques with approaches to immediately react on an observed stimulus. Although research communities are aware of the demand and it already has been part of the vision of initiatives (Kephart and Chess, 2003) a solution has not been presented yet.

One approach towards a possible solution has been described by Ye and Kalyanaraman (Ye and Kalyanaraman, 2001). They introduced an adaptive random search algorithm, which tries to combine the stochastic advantages of pure random search algorithms with threshold-based knowledge about extending the search. Their approach is based on the initial system as presented in (Ye et al., 2001). In contrast to our approach, Ye et al. propose a centralised system that tackles the optimisation task for each node. To allow for such a division of work between a central server and the particular network nodes they have to deal with problems like e. g. bandwidth usage, single point of failure, or local knowledge accessible from server-side.

3 SYSTEM

The motivation to develop a dynamically adapting system has been formulated before (cf. AC (Kephart and Chess, 2003) or Autonomic Networking (Jennings et al., 2007)), a proof of concept is still missing. The system presented in this paper is a first step towards a possible realisation. Based upon our architecture as pictured in Fig. 1 and initially presented in (Tomforde et al., 2009) the responsibilities of parameter set generation and on-line adaptation of the control system are assigned to different layers. One component (Layer 2) evolves new parameter sets not being restricted by real-time requirements for time and computation power. The other part reacts on changing stimuli (observed situation). This division of functionalities leads to the possibility that for an observed situation no matching optimised parameter set is available. In this case a covering mechanism has to cope with the situation which chooses the best possible control and adaptation strategy.

Technically, the architecture is realised by two connected techniques. As described in e. g. (Schmid et al., 2006), a combination of evolution and learning seems to be a promising solution to realise dynamic system-adaptations. Based on this assumption, for the off-line part an Evolutionary Algorithm (EA) is used in combination with a standard network simulation tool. The on-line learning mechanism is based on a modified version of Wilson's Learning Classifier System XCS (Wilson, 1995).

Within this Section the goals of the system are defined, followed by a short introduction of the basic concepts. The main part is dealing with the architecture and the current status.

3.1 Goal Definition

The network control system as presented in this paper aims at increasing the performance of data communication. It allows for the dynamic adaptation of network protocols to a continuously changing environment. Based on the initially introduced concept (Tomforde et al., 2009), the system requires *organic* (in terms of OC) characteristics – a decentral, self-organised approach leads to a stable, reliable control, the system is able to learn and optimise its behaviour autonomously. The network control system is generic, which means the controlled network protocol client can be exchanged. It supports a large set of different protocol types (Peer-to-Peer, mobile ad-hoc, wire-based, sensor, etc.) and protocols (e. g. BitTorrent (Cohen, 2003), Hyper-Gossiping (Khelil et al., 2007), etc.). If possible, the autonomous network control systems can collaborate and fullfill system-wide goals by using local interactions. The goal definition has some similarities with the manifest of Autonomic Computing (Kephart and Chess, 2003) and the Autonomic Management of Networks approach as presented in (Jennings et al., 2007). In contrast to the network-wide approach (which is unfeasible for large networks like the Internet), we assume that a locally organised solution based on local rules and local interactions will converge to a system-wide optimisation in most of the cases and therefore does not need global knowledge and global control.

3.2 Basic Concepts

The architecture is based on two already known approaches: the *Generic Observer/Controller Architecture* (Richter et al., 2006) and the *2-layered Architecture* of the Organic Traffic Control (OTC) System (Prothmann et al., 2008). The generic architecture describes an approach, where a decentralised system (System under Observation/Control (SuOC)) is wrapped with an additional surveillance and feedback mechanism. Sensors and actuators are used to monitor and control the SuOC by establishing a control loop. This control loop observes the behaviour of the SuOC through sensors, compares the results with expected behaviour and the current goals of the system, decides what action is necessary and controls the SuOC with the best known action through actuators. Additionally, a memory function keeps track of historical situations and control actions to be able to optimise the behaviour from existing knowledge.

The architecture of the OTC system is based on this approach. The realisation of a real-world scenario (control of traffic lights at urban intersections) leads to some restrictions – the main aspect is that the system has to use only parameter sets with guaranteed performance. Therefore, the situation-dependent creation of new parameter sets has been assigned to a new layer within the architecture where a simulation-based approach is performed off-line. This concept can be also found in the network control system, but the different domain leads to some modifications and changes, which will be explained in the remainder of this section.

3.3 Architecture

Similar to the *2-layered Architecture* of the Organic Traffic Control (OTC) System (Prothmann et al., 2008) our architecture consists of three parts: the SuOC at Layer 0, an on-line adaptation mechanism at Layer 1, and an off-line learning component at Layer 2 (see Fig. 1). All three layers will be presented in the following.

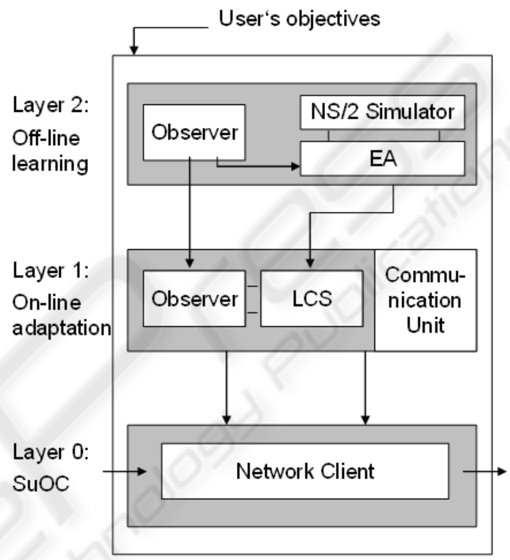

Figure 1: System architecture.

Layer 0: System under Observation and Control

The SuOC is a parametrisable Network Controller. Due to the generic concept of the proposed system it is not restricted to a particular set of protocols – the only restriction applied is that it has to provide a set of variable parameters and a local quality criterion (e. g. duration of a download in Peer-to-Peer systems or a weighted trade-off between energy consumption and broadcast-covering for MANETs) for the performance measurement. This means, protocols on all layers (media access to application) can be controlled in the same way as e. g. wire-based protocols or mobile ad hoc networks. A good setup of the variable parameters that match the current condition at the network node has an important influence on the resulting performance for these systems. In the architecture, the parameter setup is optimised on-line by the O/C component in Layer 1.

Layer 1: On-line Adaptation

The Layer 1 component can be divided into two different parts: an Observer and a Controller. The Observer is responsible for monitoring the situation at a particular node. It measures those attributes having influence on the selection of appropriate parameters for the control strategy. This selection depends on the specific controlled protocol and typically contains attributes like buffer sizes, delay times, etc. Additionally, protocol-specific parameters like e. g. number of nodes in sending distance for MANET protocols or available system resources like CPU, upload-bandwidth, download-bandwidth, etc. for P2P protocols can be taken into account. Afterwards, these values are aggregated to an abstract situation description realised as an n-dimensional vector with n equal to the number of observed values.

The main part of the Controller is a Learning Classifier System, which is based on Wilson's XCS as introduced in (Wilson, 1995). The LCS maps the aggregated input-information from the Observer to a rule base of possible actions, the process is realised in accordance with Wilson's approach. The basic change in concept is that our LCS version is not able to create new rules as this process can lead to unwanted behaviour (random rule generation). This leads to the problem that the system might not have a matching rule for the currently observed situation, although the system detects the demand to adapt the network client. Therefore, a covering mechanism is needed, which chooses the best possible action.

This covering is realised based on the assumption that a classifier whose condition part is located close to the current situation description – although it does not match it – is better than any other one existing within the rule set. Due to this assumption a covering process is executed which selects the "nearest" classifier in terms of the Euclidian Distance calculated for the n dimensional vectors (equal to the situation description) and using the centroids of the intervals used for each interval predicate. This classifier is copied, its condition part is adapted to the current situation description (using a standard interval size around the given situation), and it is added to the rule set. Based on this simple process we ensure to only use tested actions and we also ensure that at least one rule is contained in the match set. Further details on the process can be found in (Tomforde et al., 2009).

Layer 2: Off-line Learning

The existing set of classifiers and consequently the set of existing parameter sets has to be extended for situations where no classifier matches. This means, the system has to be able to autonomously learn parameter sets for unforeseen situations. Within our architecture, the Layer 2-component is responsible for this task. This component consists of three parts: an Observer, an EA and a simulator. The Observer is responsible for capturing the current situation description provided by the Observer on Layer 1. As the current usage of system components (in terms of CPU, RAM, etc.) has influence on the selection process of the LCS, the Observer is responsible for the scheduling of optimisation tasks.

The EA is responsible for evolving new classifiers. The algorithm is implemented as a standard Genetic Algorithm (cf. e. g. (Bäck and Schwefel, 1996) for details). This algorithm needs a possibility to analyse the performance of the current parameter set, which is done by using the standard network simulation tool *NS/2* (Web, 2009). The simulator needs a scenario and an implementation of the current protocol. The implementation is mandatory, but the configuration of the simulator depends on the observed situation as measured by the Observer on Layer 1. Therefore, a scenario is computed taking into account all observed attributes (e. g. for BitTorrent: number of peers, seeds, download and upload speeds, etc.)

4 RESEARCH ROADMAP

Within the previous Section the architecture of the system has been described. The system based on this has been realised and applied to a first protocol (BitTorrent - cf. (Cohen, 2003)). We demonstrated the potential of our system and validated the feasibility of our approach for a BitTorrent-based test scenario, leading to an increase in terms of the objective function (amount of downloaded data or download-time) of up to 20% (Tomforde et al., 2009). Further evaluation of this protocol is in progress. Additionally, we are working on demonstrating the applicability of our approach to other systems by replacing the SuOC (adapt and optimise mobile ad-hoc network protocols instead of a BitTorrent Client).

To completely achieve the goal as defined in Section 3.1 essential parts are still not investigated. The following part of this Section will emphasise the main focus of the future research based on this system. Therefore, we introduce the main research topics in accordance to the particular layers of our architecture.

4.1 Layer 0

The system aims at being generic in terms of controlling different protocols and protocol types. These protocols are situated at Layer 0 of our architecture as depicted in Fig. 1. To demonstrate the generic character of our approach we are going to apply the system to exemplary representatives of different protocol types.

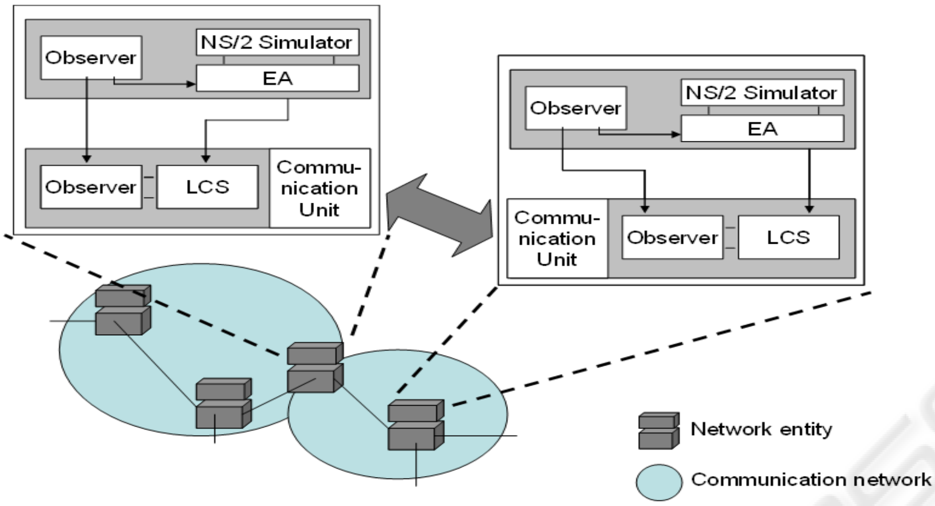

Figure 2: Collaborating network control systems.

Starting with BitTorrent as representative for Peer-to-Peer systems and the current application to mobile ad-hoc networks we will investigate the control of other protocol domains. Therefore, protocols for sensor networks are from interest as well as classical Internet protocols (like TCP/IP) and very specialised approaches like e. g. protocols for the communication in smart camera networks (Hoffmann et al., 2008).

In addition to the application of different protocol types, we aim at extending the control scope to *cross-layer optimisation* (Wang et al., 2005). This means e. g. for TCP/IP that the configuration of IP is selected depending on the current situation and of TCP depending on the configuration of IP.

4.2 Layer 1

The performance of the on-line adaptation mechanism at Layer 1 depends primarily on the applied learning technique. Due to this dependency we aim at validating the usage of our LCS by comparing it to other learning techniques. Therefore, existing techniques will be analysed based on the usage within our architecture and implemented if promising.

In addition to the increase of performance by analysing the learning component, the overall performance can be increased by allowing for collaboration. Neighboured entities should get the ability to collaborate with each other (see Fig. 2) in order to schedule Layer 2 tasks, exchange knowledge, and avoid redundant simulation-based learning.

4.3 Layer 2

As the off-line generation of new parameter sets is resource- and time-consuming, an improvement is necessary. The urgent target for this component is to speed up the rule creation process. The approach to solve this problem consists of two different strategies: a speed-up at start time and an approximation at runtime. At start time the system does not have any other rules than the standard parameter set of the protocol, which leads to the need of a fast mechanism to learn rules for a set of exemplary rules within the configuration space. These rules might have a lower quality than an optimised one, but they will be replaced with an optimised version during runtime.

The other aspect of the speed-up process at Layer 2 is, that nodes might not have sufficient resources for the optimisation task (e. g. sensor networks). Hence, research here will focus on approximating a reasonable parameter set (taking those parameter sets into account which are situated close to the situation description). This means, an intelligent inter- and extrapolation mechanism is needed in combination with a half-centralised solution (e. g. periodic updates from a service running on other nodes).

Another aspect of the Layer 2 optimisation is to use stand-by time (no active optimisation task) to optimise the coverage of the configuration space. This means some kind of active learning may be used to pro-actively generate parameter sets for situations where currently no adequate parameter set is known. Additionally, a collaboration mechanism will be helpful to schedule these active learning tasks for a set of neighboured entities.

5 CONCLUSIONS

This paper presented a system for the dynamic adaptation of network protocol parameters. The system monitors the situation at particular nodes and reacts on changes by adapting the communication protocol client. It is able to learn new control strategies and works on a self-organised basis. We explained our position that the presented system will be able to increase the performance of future communication networks without changing the whole technical background. Finally, we named the main research fields for our approach based on the introduced goal. Our system can also serve as a good testbed for the investigation of innate aspects of OC systems like trustworthiness or collaboration patterns.

REFERENCES

Bäck, T. and Schwefel, H.-P. (1996). Evolutionary computing: An overview. In *Proceedings of IEEE Conference of Evolutionary Computing*.

Cohen, B. (2003). Incentives Build Robustness in BitTorrent. In *Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems, Berkeley*.

Handley, M. (2006). Why the internet only just works. *BT Technology Journal*, 24(3):119–129.

Hoffmann, M., Wittke, M., Bernard, Y., Soleymani, R., and Hahner, J. (2008). Dmctrac: Distributed multi camera tracking. *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on*, pages 1–10.

Jennings, B., van der Meer, S., Balasubramaniam, S., Botvich, D., Foghlu, M. O., Donnelly, W., and Strassner, J. (2007). Towards autonomic management of communications networks. *Communications Magazine, IEEE*, 45(10):112–121.

Kephart, J. O. and Chess, D. M. (2003). The Vision of Autonomic Computing. *IEEE Computer*, 36(1):41–50.

Khelil, A., Marron, P. J., Becker, C., and Rothermel, K. (2007). Hypergossiping: A generalized broadcast strategy for mobile ad hoc networks. *Ad Hoc Networks*, 5:531–546.

Montana, D. and Redi, J. (2005). Optimizing parameters of a mobile ad hoc network protocol with a genetic algorithm. In *GECCO '05: Proc. of the 2005 conference on Genetic and evolutionary computation*, pages 1993–1998, New York, NY, USA. ACM.

Prothmann, H., Rochner, F., Tomforde, S., Branke, J., Müller-Schloer, C., and Schmeck, H. (2008). Organic control of traffic lights. In *Proc. of the 5th Intern. Conference on Autonomic and Trusted Computing*.

Richter, U., Mnif, M., Branke, J., Müller-Schloer, C., and Schmeck, H. (2006). Towards a generic observer/controller architecture for Organic Computing. In Hochberger, C. and Liskowsky, R., editors, *INFORMATIK 2006 – Informatik für Menschen!*, volume P-93 of *GI-Edition – Lecture Notes in Informatics (LNI)*, pages 112–119. Köllen Verlag.

Schmeck, H. (2005). Organic Computing – A new vision for distributed embedded systems. In *Proceedings of the 8th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'05)*, pages 201–203.

Schmid, S., Sifalakis, M., and Hutchison, D. (2006). Towards autonomic networks. In *3rd Intern. Annual Conference on Autonomic Networking, Autonomic Communication Workshop (IFIP)*, Lecture Notes in Computer Science. Springer Verlag, Heidelberg.

Siekkinen, M., Goebel, V., Plagemann, T., Skevik, K.-A., Banfield, M., and Brusic, I. (2007). Beyond the future internet–requirements of autonomic networking architectures to address long term future networking challenges. *Future Trends of Distributed Computing Systems, IEEE International Workshop*, 0:89–98.

Sözer, E. M., Stojanovic, M., and Proakis, J. G. (2000). Initialization and routing optimization for ad-hoc underwater acoustic networks. In *Proc. of Opnetwork'00*.

Tomforde, S., Steffen, M., Hähner, J., and Müller-Schloer, C. (2009). Towards an organic network control system. *submitted for publication*.

Turgut, D., Daz, S., Elmasri, R., and Turgut, B. (2002). Optimizing clustering algorithm in mobile ad hoc networks using genetic algorithmic approach. In *Proc. of the IEEE Global Telecommunications Conference (GLOBECOM '02)*, pages 62 – 66.

Wang, J., Li, L., Low, S. H., and Doyle, J. C. (2005). Cross-layer optimization in TCP/IP networks. *IEEE/ACM Trans. Netw.*, 13(3):582–595.

Web (2009). The Network Simulator - NS/2. *http://www.isi.edu/nsnam/ns/*.

Wilson, S. W. (1995). Classifier fitness based on accuracy. *Evolutionary Computation*, 3(2):149–175.

Ye, T., Harrison, D., Mo, B., Sikdar, B., Kaur, H. T., Kalyanaraman, S., Szymanski, B., and Vastola, K. (2001). Network Management and Control Using Collaborative On-line Simulation. In *Proceedings of IEEE ICC*, Helsinki, Finland. IEEE.

Ye, T. and Kalyanaraman, S. (2001). An adaptive random search algorithm for optimizing network protocol parameters. Technical report, Rensselaer Polytechnic Inst.