Demo Abstract: Cross-layer optimization of edge networks

Elena Meshkova, Andreas Achtzehn, Janne Riihijarvi and Petri Mahonen*
Department of Wireless Networks, RWTH Aachen University
Email: {eme, aac, jar, pma}@mobnets.rwth-aachen.de

ABSTRACT

In this demonstration paper we suggest an architecture and provide a prototype implementation of a *cross-layer and cross-network* optimization engine that aims for centralized *autonomous management* of edge networks using *network utility maximization* approach [2]. We focus on these networks as many performance bottlenecks are encountered there, especially in the wireless domain. Our approach relies on abstraction of a network as a number of interacting services for which user-defined objectives are expressed using *utility functions*. We model a network as a "black box" with known adjustable inputs and measurable outputs. To optimize such a system we construct a feedback loop based on a *simulated annealing* algorithm to achieve a fast convergence to a near-optimum result, and employ *approximate probabilistic graphical models* for re-use of previously gathered data. We demonstrate on examples of scenarios executed in both a wireless testbed and a network simulator that such an optimizer enables effective autonomous cross-layer network stack parameter optimization.

Categories and Subject Descriptors: C.2.1 Computer-Communication Networks – *Wireless communication*
General terms: Optimization, network management
Keywords: Wireless networks, cross-layer optimization, simulated annealing, approximate graphical models

1. SYSTEM ARCHITECTURE

The architecture of the autonomous network optimization platform is based on the definitions of the autonomic control loop and the intelligent agent [3] (see Figure 1). The optimizer, called the *cognitive engine* (CE), considers the network as a "black box" that hosts services, which can be provided, for example, by individual protocols, nodes or even whole networks. A user formulates her needs in terms of *objectives* that a set of services should achieve. They are stated as a *utility function* that is further restricted by a number of *policies*. Most of the networking objectives today such as QoS classes, elastic bandwidth-sharing typical to file-transfer application, and economical considerations, like access costs, can be understood as utilities (see Figure 2.a-c).

A utility function depends on measurable characteristics of services, called *attributes* ("knobs" or "sensors" in other literature). Services can be adjusted using *parameters* ("dials" or "actuators"). A change in parameter settings results in different attribute readings and therefore affects the utility. Examples of parameters are TCP congestion avoidance algorithms or a contention window size of CSMA/CA MAC protocol. Attributes might include delay, throughput, and usage fees. The goal is to find a subset of services and their parameter configurations, so that the user-defined utility is maximized. In general it is an NP-hard problem.

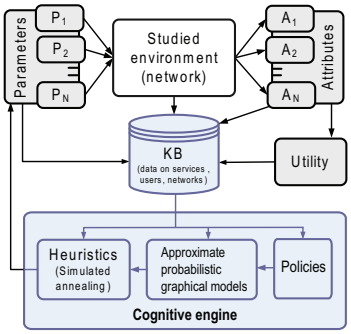

Figure 1: Architecture of the auto-configuration engine.

The functional core of the CE is a *simulated annealing* (SA) algorithm that is known to be effective for "black box"-type of problems [3]. Typically it can achieve a near-optimum stable solution with a small number of iterations. The *search space* for SA is a multi-dimensional landscape where each dimension corresponds to one adjustable parameter. The CE iteratively acquires the parameters' states and estimates their influence on the network performance. If needed the engine tunes the parameters. The process is repeated until the best possible stable solution is reached, and re-initiated if the performance degrades or at regular intervals in order to detect an improvement in network conditions.

We have also added reinforcement learning techniques to SA that boost on-line learning capabilities of the basic algorithm [3]. In order to represent dependencies learned by the optimizer, which can be used either to give "direction" to the search or change the search space, we have considered a number of *approximate probabilistic graphical models*.

2. DEMONSTRATION DESCRIPTION

We demonstrate the performance of the cognitive engine on two different scenarios. The first scenario is set in a wireless testbed with four wireless clients and an access point. The second scenario focuses on multiple wireless accesses

*Authors would like to thank RWTH Aachen University and EU (Aragorn-project) for financial support.

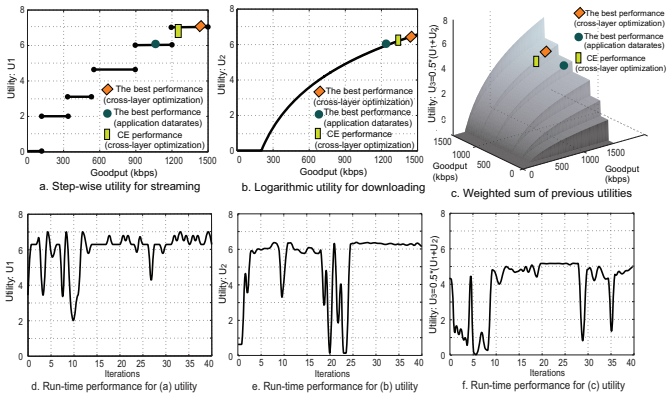

Figure 2: The performance of the cognitive engine over a combination of utility functions. Figures (a)-(c) show the average utility values. Figures (d)-(f) show run-time performance of the CE.

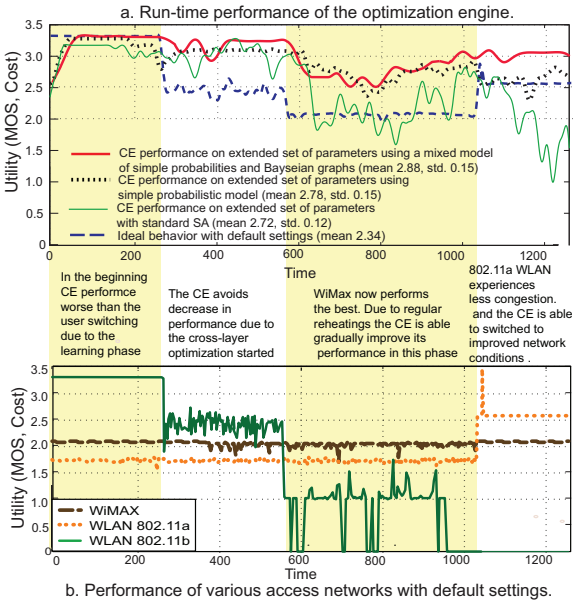

Figure 3: The time-evaluation of performance of the VoIP scenario showing learning capabilities of the system and optimization gains.

and is run in Qualnet [1]. The cognitive engine is realized in MATLAB. It interacts with the testbed or the simulator getting performance statistics and setting parameter values as result of the optimization process.

2.1 Simple testbed scenario

We have four wireless clients connected to one 802.11g access point. First two clients download data over TCP with logarithmic utility. Other two are running the streaming application over UDP with step-wise utility (see Figure 2.a-c). The parameter values of each client are determined centrally by the CE residing on an external machine. The purpose of the scenario is to rapidly achieve maximum global utility.

The performance results of the CE are given in Figure 2. We test three combinations of utilities: optimizing separately for the streaming and downloading applications and for a combination of both. The optimized parameters are application datarates, MTU, TCP congestion algorithm, RTS /CTS threshold and the MAC's PDU, resulting in around 2,500 permutations of parameter values.

As shown in Figure 2 the optimizer is capable to autonomously achieve the performance close to the absolute maximum found by the exhaustive search through all parameter combinations. Moreover, in case of the step and the mixed utilities it obtains higher gains than achieved by sole adjustment of application datarates with all other parameters kept at the default levels by 13.4%. This shows the validity of autonomous cross-layer optimization in the wireless environment. We also observe the fast convergence of the CE which on average takes only 15-25 iterations.

2.2 VoIP traffic

In this scenario the user initiates a VoIP call. The utility function is a combination of the Mean Opinion Score (MOS) and a cost function. The user has access to WLAN 802.11g, 802.11a and WiMAX networks. The Wi-Fi access points are free of charge, but over time they become saturated, hence their performance decreases (see Figure 3.b). The user has to pay for WiMAX usage, but the capacity of this network satisfies all her requirements and the user switches to it only when the MOS drops low. We assume seamless vertical handovers in this scenario. The ideal device behavior in terms of interface switching, with protocols parameters kept at default levels, are shown by the dashed line in Figure 3.a.

Example results have shown that autonomous switching of access methods performs only slightly worse than the manual one with average performance degradation not exceeding 4.2%. However, when the engine can additionally adjust the MAC protocol settings and the MTU size, the average gain compared to the default scenario becomes 23.2% (see Figure 3). This happens because cross-layer optimization allows the user to have a reasonable MOS in a cost-free Wi-Fi network and switching to WiMAX is avoided.

Additionally we have evaluated the performance of the CE with three approximate graphical models besides the basic SA algorithm. The first is based on *marginal probabilistic approach*, storing probabilities if a decrease or increase of a certain parameters leads to the change in the utility. The second is based on *Bayesian networks* and the third is a combination of the above. The results (see Figure 3) delivered by these models are promising and we hope they can raise some discussion during the demonstration session.

3. REFERENCES

[1] QualNet. Scalable Network Technologies, http://www.scalable-networks.com [last visited on 23.04.2009].

[2] F. Kelly, A. Maulloo, and D. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research society*, 49(3):237–252, 1998.

[3] S. Russel and P. Norvig. *Artificial Intelligence A Modern Approach*. Pearson Education, Upper Saddle river, New Jersey, USA, 2003.