Towards Self-optimizing Protocol Stack for Autonomic Communication: Initial Experience

Xiaoyuan Gu[1], Xiaoming Fu[2], Hannes Tschofenig[3], and Lars Wolf[1]

[1] Institute of Operating Systems & Computer Networks,
Technische Universität Braunschweig
Mühlenpfordtstr. 23, 38106 Braunschweig, Germany
{xiaogu,wolf}@ibr.cs.tu-bs.de
[2] Institute for Informatics,
Universität Göttingen
Lotzestr. 16-18, 37083 Göttingen, Germany
fu@cs.uni-goettingen.de
[3] Siemens AG
Otto-Hahn-Ring 6, 81739 Munich, Germany
Hannes.Tschofenig@siemens.com

Abstract. The Internet is facing ever-increasing complexity in the construction, configuration and management of heterogeneous networks. New communication paradigms are undermining its original design principles. The mobile Internet demands a level of optimum that is hard to achieve with a strictly-layered protocol stack. Questioning if layering is still an adequate foundation for autonomic protocol stack design, we study the state-of-the-art from both the layered camp and its counterpart. We then outline our vision on protocol stack design for autonomic communication with the POEM model and its internals. A novel cross-layer design approach that combines the advantages of layering and the benefits of holistic and systematic cross-layer optimization is at the core of this work. With inspirations from the natural ecosystem, we are working on the role-based Composable Functional System for self-optimization that features proactive monitoring and control. By doing so step-by-step, we envisage reaching the goal of self-tuning autonomic network with high level of autonomy and efficiency, with minimum human management complexity and user intervention.

1 Introduction

What is the Internet? Is it a technology, an industry, a communication medium, or a kind of society? The Internet is all of these and none of these. It is an ecological system - the Internet Ecosystem, and like all the ecosystems it grows, spawns, may be attacked, builds up and declines. Yet, it is extremely complex. Complexity sources from its infrastructure, network management, heterogeneity in devices and access schemes, abundant services and applications. Complexity is amplified by the speed at which the Internet evolves both technologically and in population. With the worldwide wireless buildout, isolations between different communication systems are diminishing. The trend of everything over IP and IP connects everything is pushing all kinds of networks, wired or wireless, towards integration, composition and interworking.

While the users are benefiting from emerging technologies and convenience, the operators suffer from looming complexity in the construction, configuration and management of such networks. The traditional way of manual planning, configuration, trouble-shooting, policy making and optimization will be exorbitantly expensive or even dominate operational cost, as opposite to hardware/software improvements that continuously help to reduce capital expense. Increasing size of the network infrastructure and shortage of skilled labor for the management of complex systems further convolutes this crisis. In one word, the extent of complexity may eventually exceed the capability of human being, and undermine reliability and end-user trust of the system.

Managing complexity is not the only concern of today's Internet. Rethinking of its design principles represents another urgent agenda. Dated back to the 70's, the early Internet was designed with strict layering and an end-to-end model for its architecture, which was not able to foresee today's pervasive middlebox communications. middleboxes like firewalls, NAT boxes, proxies, explicit/implicit caches basically break the original end-to-end arguments. Other multi-way interactions such as QoS, multicast, overlay routing, and tunneling also contribute to the violations on the layered model. Emergencies of sublayer technologies like TLS at layer 4.5, IPsec at layer 3.5, MPLS at layer 2.5, and wireless networks specific sub-link layers (e.g. RLC, RRC, PDCP) stir up the trouble. To sum it up, the complicated interactions make it difficult to describe using strict layering, and layering often lets some new services fit poorly into the legacy structure.

Apart from the wired network domain, the recent advances in the wireless communications have raised architectural concerns from another perspective. Traffic variability, topology dynamicity, heterogeneity in access technologies, constraints like radio resource, energy in 3G/4G mobile networks, wireless LANs, Mobile Ad Hoc Networks, Micro Sensor Networks, DVB-H Networks, and QoS in real-time interactive mobile multimedia applications, are putting traditional design methodology on protocol stack under examination. A common understanding here is that traditional layering is the source of most performance related problems, and shared information among the protocols layers is critical for performance optimization in wireless networks and the Mobile Internet. With the world-wide push of the wireless communications towards an All-IP infrastructure, the issue of a good architecture is ever more important.

However, giving up layering is extremely difficult, as layering is a natural way of dealing with complex systems. The huge success of the Internet is to a great extent due to its layered architecture. By organizing the communication functions into hierarchical and nested levels of abstractions - the protocols layers, modularity and open interfaces are ensured. This simplified the development of networking protocols and applications, and hence the proliferation of the Internet.

So, the obvious question now is: what would be the right way of structuring the communication software - the protocol stack? We argue that in facing the above-mentioned problems, firstly, there is a need to make future networks self-govern, in the sense that it works in an optimal way with endogenous management and control, and with minimum human perception and intervention. Secondly, a trade-off between architecture and performance has to be in place, and likely a solution for this would be a hybrid architecture that combines the layering for the basic functionalities of the protocol stack, and a non-layered approached for performance-oriented control plane. Such paradigm allows managing complexity, will be better compatible with middlebox communication, and will fulfill the performance requirements in the Mobile Internet.

The rest of the paper is structured as follows. We study the state-of-the-art in the research on network architecture and protocol stack design both from the layered camp and its counterpart in Section 2. This is followed by our vision on the architecture for autonomic communication protocol stack in Section 3, as detailed by the Performance-oriented Reference Model, the AutoComm protocol stack design and prototyping, workflow of self-optimization, as well as the determination of critical control points. Finally we conclude our studies and outline the directions for future research in Section 4.

2 Related Work

2.1 Autonomic Computing

Autonomic computing [1] has in the past few years attracted pretty much attention as a novel computing paradigm. Not only being an area of intensive research in academia, Autonomic Computing has also become a strategic goal of prominent IT companies like IBM, Sun, DaimlerChrysler and Fujitsu-Siemens [2]. Basically, it is a concept of self-managed computing systems with minimum human conscious awareness or involvement, derived from the human autonomic nervous system - a sophiscated computing device and autonomic entity. Still in its early stage, to date, most work on autonomic computing can find its source from neurosciences and biology. In [3], the essence of autonomic computing, architectural considerations, engineering and scientific challenges are thoroughly analyzed. Opportunities and possible research directions of autonomic computing in the system engineering field are well explained in [4]. A bottom-up approach in system design for effective emergency control and handling using so called Observer/Controller architectures is proposed in [5]. Self-organization, self-adaptivity, reconfigurability, and emergence of new properties are topics in a variety of research projects in fields like middleware [6, 7], database system [8], and software engineering [9, 10]. Cisco, together with IBM is proposing a service framework [11] consisting of a set of potential interface specifications for adaptive remote service and support systems, which enables the customers to interact with the ISPs for autonomic detection, diagnosis, and rectification.

2.2 Autonomic Communication

Despite the heat in the computing area, it is until recently that seeking technical usages of principles observed in natural systems in communication arena has been undertaken. The newly founded Autonomic Communication Forum [12] and its initiative [13] are becoming a call to arms for concerted intellectual efforts towards next generation telecommunication. The University of Bologna is building a framework [14] to support the design, implementation and evaluation of peer-to-peer Internet applications using Swarm Intelligence. A number of projects [15] are going on within the scope of bio-inspired (e.g. from bacteria) approaches for autonomous configuration of distributed systems at the University College London. Based on a chemical reaction model, a new approach [16] with the concept of fraglets for self-healing communication protocol stack has been proposed by University of Basel. All of these efforts hinges on a central theme: autonomic communication.

Autonomic Communication (called AutoComm here after) [17] treats the Internet as an ecosystem - the Internet Ecosystem. By definition, AutoComm represents the study of the inter-relationship between networks or network elements and their situations from a cross-disciplinary perspective, and a methodology of using context-awareness and distributed policy-based control to achieve efficiency, resilience, immunity and evolvability in large-scale heterogeneous communication infrastructure. AutoComm focuses on populations, not individuals, and it seeks balance and optimization on the dynamics of the relationship. A key element in AutoComm is the situation, or called context, which can be understood as a capture for a multi-faceted, uncertain and varying set of communication purposes, policies, conditions, requirements, states, etc. from regulatory, social and private down to technical and engineering. It is of vital importance for AutoComm to understand how network elements behaviors are learned, influenced and modified, how these affect other elements, groups and networks, and how these can offer purposeful inputs on deciding the design principles of the network architecture and protocol stack. The ultimate contribution of AutoComm R&D will be to enable an evolving network platform for sensing, communicating, decision making, and reacting, with high degree of autonomy to ease human efforts and high level of management efficiency in the Operations Support Systems (OSS) in the Telecom industry.

2.3 Cross-layer Design

Cross-layer Design shares the same motivation of optimal performance of the Mobile Internet as AutoComm. Mobile and wireless networks have a number of characteristics that differentiate them from their wired counterparts, for which one has to think twice before simply borrowing the recipe of the success of Internet and applying its architecture to mobile and wireless networks.

One obvious shortcoming of the two classical models - OSI Reference Model and TCP/IP Model is the lack of information sharing among the protocol layers [18]. This hampers optimal performance of the networks due to the fact that shared layer information is the prerequisite for performance optimization. *Cross-layer design* represents a violation of over-strict layering and too tightly controlled interactions, by encouraging better communications between the protocol layers with holistic and systematic methodology to improve overall system performance.

To date, most existing cross-layer design approaches to a large extent focus on direct interactions between the protocol layers by involving only two or three layers and dragging shortcuts between protocols [19, 20]. Cross-layer design is no easy task, as the cooperation among multiple protocol layers has to be coordinated without endangering conflicts and loops. A common drawback of the current approaches is missing a holistic approach for cross-layer design (not just interactions). Furthermore, once the layering is broken, the luxury of designing protocols in isolation is lost. Also, unbridled cross-layer interactions can create loops, and from control theory's point of view, they become hazards to the stability of the system. Loosely-controlled interactions can also result in "spaghetti code", which basically stifles further innovation and proliferation on the one hand, and increases the cost for upkeep on the other hand. In severe cases, the overall system will have to be redesigned should some key modules change in the future. These problems are detailed in [21] with live examples as proofs.

2.4 Protocol Heap and Role-based Architecture

If Cross-layer Design is considered as renovation to the architecture of the current Internet, some of the approaches are heading for revolutions - to change Internet's architecture thoroughly by totally giving up layering. Role Based Architecture (RBA) [22] and its Protocol Heap is a good example. Being an ongoing DARPA funded effort toward a new architecture for next generation Internet, it aims to replace layering by roles that correspond to individual communication building blocks. As can be seen from Fig. 1, an arbitrary collection of sub-headers from conventional protocols headers are used to form role data - the Role Specific Headers (RSHs). They are then structured as heap rather than stack to serve as packet headers. RSHs can be added, modified or deleted along the forwarding path.

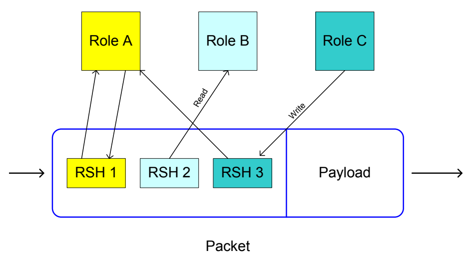

Fig. 1. Role-based Architecture

Obviously, giving up layering can have better functional modulization, flexibility, extensibility, easier in-band signaling, auditability and portability. But these do not come for free. Radical changes of a well established and highly successful architecture will cause compatibility problems. Also efficiency of processing, possible increased complexity and confusion will be questioned. The work is still in conceptual phase, awaiting realization and resolution of many open issues.

2.5 Multi-domain Communication Model

Criticizing that protocol stacks are architecturally static and not knowing where the communication is heading for, the Multi-domain Communication Model (MDCM) by Wang et al. [23] proposes to use domains to organize communication building blocks. By concept, domain is a logical construct of the common protocol layers in individual stacks along the communication pathway. Domains are defined by their specific addresses, namespaces and channel properties. Domain specific messages are encapsulated with such definition with correspondence to the protocol header. Hereby, communication can be understood as a process of recursive domain traverse and selections from end to end. Moreover, different from conventional stack approach, MDCM allows dynamic determination of the relationship between the protocol layers using pluggable functions and algorithms. The MDCM builds upon the existing stacks and integrates the next-domain(layer) determination, forwarding and resolution functions into a unified recursive model. Fig. 2 gives an example of using two domains - the IP domain and Ethernet domain to interpret the communication procedure that involves a name resolution with ARP in the LAN.

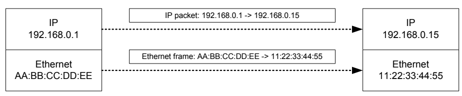

Fig. 2. Multi-domain Communication Model

Although an architecture based on this model allows more relaxed relationship and dynamic binding between the protocol layers, this is more or less a different kind of reasoning of the packet forwarding function of the stack, from a top-down view instead of a bottom-up view which is common in conventional models. What are obviously missing are the new capabilities to enable programmability, self-organization, context-awareness, high degree of autonomy and minimum human intervention, and to deal with prevalent middlebox communication.

2.6 Region-based Interworking Architecture

Compared to RBAs approach of functional-oriented granularity, the region-based work tries to divide the Internet using explicit architectural components with the concept of regions. A region in such context is a partition of the network with consistent state, knowledge and control. A collection of interconnected regions represents a connected set of heterogeneous networks. At region boundaries, special gateway or way pointing entities are adopted to facilitate identity mapping, routing information exchange, and message formats representation. Catenet [24] is such a scheme that pioneered the architecture for the Internet with descriptions and criteria in the late 70s. Wroclawski [25] defined the waypoints in his Metanet Model for the description of the transitions from one region to another. This is enhanced by the Regions Project [26] that provides a more generic mechanism for grouping, partitioning, and formalizing boundaries around the groups and partitions. Plutarch [27] is closer to the domain, but it relies on explicit state maintenance along the paths, using a principle similar to the ATMs virtual circuits. Realm-Specific IP (RSIP) [28] and 4+4 [29] went a step further by using different mechanisms to traverse heterogeneous regions.

Regions capture the partitions of homogeneity in the larger-scale heterogeneous communication infrastructure, and focus on issues like interoperability and bridging between heterogeneous networks while leaving the details of the boundary crossing embedded in the waypoints. Unfortunately, most of these work concentrated only on the forwarding function of the communication system, which is only a partial solution to the network architecture as a whole.

2.7 Non-architectural Approaches

Beside architectural approaches mentioned above, many self-optimization schemes have been proposed in recent years. Dated back to 1996, another DARPA project by Tung etc. [30] introduced how to design self-organizing agents that representing finite state automata, to work together collaboratively for maximum optimization in a distributed system. Gausemeier [31] described in a self-optimizing autonomous mechatronic system that consists of intelligent agents, sensors, actuator etc. from four perspectives: target, structure, behavior and parameters. In [32] a proactive online control technique for self-optimization in information system was proposed. The actions that govern system operations are based on optimization of forecasted system behaviors, described using a mathematic model for the specified QoS criteria over a limited look-ahead prediction horizon. Krishnamachari gave a very good overview in [33] on self-optimization in communication with the environment (e.g. sensor networks). Two important views were given. Firstly, the performance of protocol stack must be analyzed with respect to a combination of environment effects, application specifications and protocol parameters. Secondly, protocols must be designed to be self-optimizing, improving autonomously over time by incorporating sensor observations. In [34], a model using so called overall business metric (OBM) was introduced for self-optimizing resources of an IT infrastructure and keeping the infrastructure aligned with business objectives.

3 Our Approach

We consider self-optimization an endogenous process of consistently adjusting the target performance vectors on situational changes, and autonomously adapting the structure, behavior and parameters of a networked ecosystem towards optimal communication efficiency and evolvability. Such a process is a composable/composite function (CF), as can be exemplified by roles like general QoS, resource management, energy efficiency, routing, economic balance etc. Self-optimization should also involve translating business policies into technical counterparts, classifying system policies and map them to the optimization roles, enhancing those policies through learning, context-awareness and conflict resolution, as well as the self-assessment of overall performance using metrics cover both technical and business domains.

One of the enablers in AutoComm will be the innovative approach of the organization of the communication software itself - the cross-layer optimized and situation-aware protocol stack. A self-optimizing AC protocol stack in this context has to face the following challenges:

- Architectural and instrumental considerations with interfacing and compatibility to the current Internet.
- Identification and representation of individual optimization functions and their metrics.
- Dynamic composition and decomposition of self-optimization with functional roles.
- Optimization data processing regulation and execution scheduling.
- Context awareness in self-optimization.
- Distributed and proactive policy-based control in self-optimization.

In answering the challenges, we first give our vision of architectural considerations on interfacing and compatibility to the current Internet with the POEM reference model. We then address the functional considerations of self-optimization with the COP protocol.

3.1 Innovative Approach of the Organization of the Communication Software

We have been working on the Performance-Oriented Reference Model (POEM) (see Fig. 3) that incorporates AutoComm flavors. Conceptually introduced in [35], POEM has no intention to radically change the current Internet architecture by entirely giving up layering. Neither does it follow the protocol heap concept. It is a novel cross-layer design approach that combines the advantages of layering and the benefits of holistic and systematic cross-layer interactions.

The basic design criterion is self-optimization is a control plane issue, where the normal functions of the protocol stack should not be compromised, and on-top of that to put add-on benefits of controlled cross-layer optimization. As illustrated in Fig. 3, POEM is composed of two conceptual planes: the user plane for normal data flows just like without cross-layer optimization, and the control plane for optimization interaction flows between two protocol layers, between a protocol layer and optimization role specific data, as well as between-roles. The interactions are all done through the defined Common Optimization Interface (COIN). The logical Common OptimizatiOn Layer (COOL) is responsible for offering Self-Optimization Service (SOS), as implemented by its Common Optimization Protocol (COP).

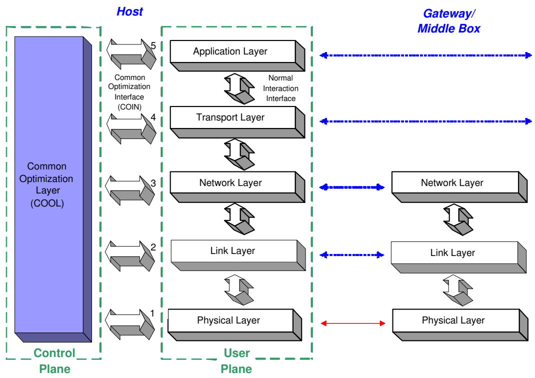

Fig. 3. Performance-oriented Reference Model

3.2 Cross-layer Optimized and Situation-aware Protocol Stack

In an AutoComm system with laws and rules that guiding its efficiency and evolvability, structural and behavioral things are best ways to express the static and dynamic features of self-optimization in the form of protocol. The COP is designated for this task. The main targets of COP are to realize context awareness in community communication, and to perform distributed policy-based control for role-based optimization composite function. Like any protocol, COP has its protocol data unit (PDU). First of all, we propose to organize the ROle-Based INformation (ROBIN) that contains role-based functional entities for stack-wide and node-wide optimization as a heap. Secondly, the conventional protocol stack is structured as a stack, which is left intact due to the reasons mentioned earlier. We then use a frame stack to control the access to the heap and the stack as depicted in Fig. 4. As can be easily understood, the frame stack and the heap are actually corresponding to the header of a COP frame. The necessary stack data of the conventional protocols headers plus the payload of a packet form the payload of a COP frame.

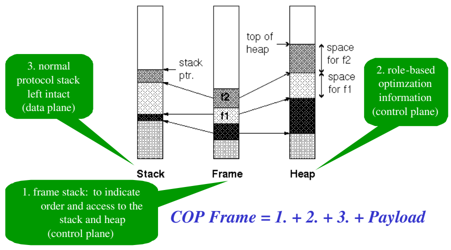

Fig. 4. Data Structure of COP Frame

3.3 Prototyping with a Natural Ecosystem

Interesting enough, we found there exists such prototype from the nature. Consider a simplified ecosystem (see Fig. 5) formed by the lion, the giraffes, the trees, as well as bacteria and fungi. The soil, the air, the sunshine, the water - all the inorganics are the data plane. The plants, animals, microorganisms - all the organisms are our control plane. The trees are sensors that use their roots and leaves to transform sunshine, water, nutrients whatever through photosynthesis into energy to feed the upper-hierarchy animals-the giraffes. The giraffes unfortunately grow up to become the meals of the lion. Noticeable, herbivores do have certain intelligence, and they are able to digest and absorb the food and convert them into flesh to serve the lions (although most likely unwillingly). The flow of energy from the plants to the herbivores and then to carnivores, is just like the way information transverses in the protocol stack, with similar entity mapping as well. On the reverse direction, all old leaves of the trees, dejection (wastes) and dead bodies of the animals are used by the bacteria and fungi - the actuators, who decompose and return some of the elements (the feedbacks) back to the earth to influence its structure. Things work out self-organized and self-optimized. If there are insufficient trees, some of the giraffes will leave or die - the balance is kept.

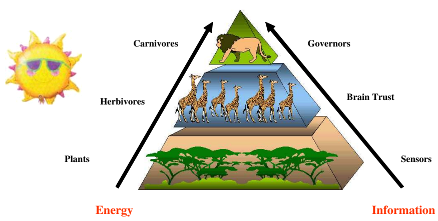

Fig. 5. Inspirations from the Natural Ecosystem

We have observed at least these from our great nature: Fist of all, it is the rule of “Natural selection” that governs the optimum operations of such ecosystem. Second, the organisms compete to survive, learn to improve, adapt to situate, evolve to prosper, or if they fail to do so, they die or extinct. Third, in doing so, they take the initiative, act proactively rather than reactively, and they often make good use of their environment - the situation, to help to adjust their behaviors. Forth, the intelligence of the organism increases while going up the food chain, the same for the density of the energy contained in the food as more and more processing is involved. Fifth, the consumption of the energy mimics a "pull" mode rather than "push". All of these have motivated us to put more efforts on the inter-disciplinary studies of the natural principles, to extract inspirations and use them to form the foundation for the research in AutoComm protocol stack.

3.4 Matching to Self-optimization in AutoComm

To apply the above paradigm to self-optimization in AutoComm, at layers and sub-layers of normal protocol stack that are relevant to optimization, critical control points (CCP) are set and sensors are correlated for the aggregation of stack-wide context. Sensors are also spread to sense the network elements environment to help to generate and update node-wide and network-wide context. These sub-contexts are then used to form the Common Optimization COntext (COCO). COCO is the basis for carrying out prediction, analysis, learning, conflict resolution, decision and action that are part of policy-based control. This is the task of the Brain Trust as illustrated in Fig. 6. The overall behaviors of a self-optimization function are coordinated by a governor, who is responsible for a number of tasks like translating business policies into technical counterparts, classifying system policies and map them to the optimization role, producing optimization performance metrics (see Fig. 8) that cover both technical and business domains, as well as the self-assessment of overall performance based on the metric(s).

3.5 Self-Optimization with Role-based Composable Function System

We consider a self-optimizing AutoComm System a Composable Functional System (CFS), in which individual optimization functions, the components of such system, can be composed and reconfigured according to needs. This envisions flexibility, extensibility, and evolvability - design for yet unknown. As depicted in Fig. 7, identifying the application domain represents the starting point of the workflow of such system. For example, a domain can be either network management, or network planning, or QoS provisioning, or multimedia service composition - you name it. Use case analysis and the formation of the performance evaluation metrics add to the initial step with greater details. Here the use cases are adopted to capture the intended behavior of the CFS, without having to specify its internal implementation. Requirements are captured, illustrated and implied to help system's end users and domain experts to reach a common understanding. Furthermore, use cases serve to validate the system's architecture and

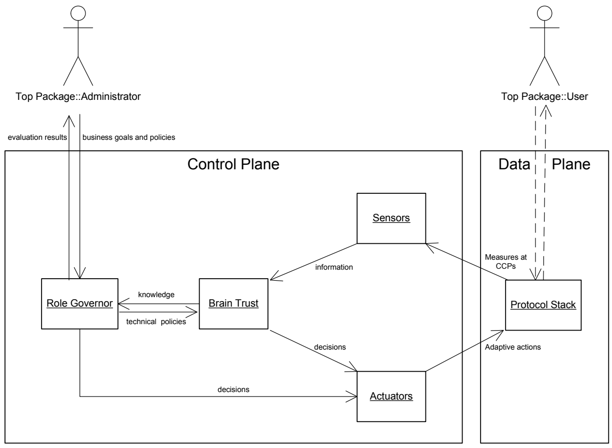

Fig. 6. Interactions Among the Self-optimization Entities

to verify the system as it evolves. To give a few examples: delay optimization, jitter optimization, loss rate optimization, bandwidth consumption optimization, energy consumption optimization, radio resource optimization, processing overhead optimization, storage capacity optimization, financial cost optimization and so on. Each of the use cases can be further divided into sub-use cases, depends on the level of granularity.

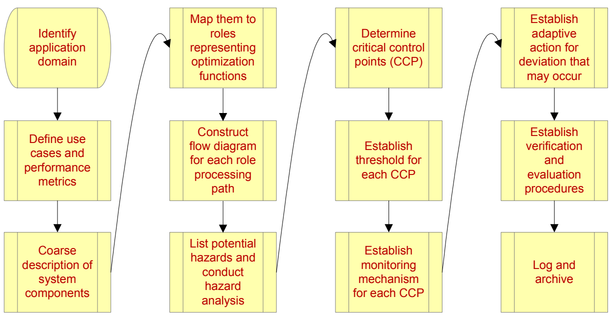

Fig. 7. Sequence Diagram of Role-based Self-optimization

Associated with the use cases are the performance metrics for individual functional components, established by mapping the business objectives and policies (e.g. Service Level Agreements), to technical qualitative and quantitative measures (e.g. QoS parameters). This is then followed by the construction of the CFS with coarse description of the system functional elements. Here the static things - the entities and their relationship, and the dynamic things - the activities and interactions among the entities (such as that depicted in Fig. 6 above) are at the core of the work. The use cases are then mapped to the roles that each represents a specific aspect of the composite optimization function. All the related processing routines (can be either for the end-system only, or end-to-end across the network) for a role are described with its own flow diagram afterwards. Potential hazards (factors that will have negative impacts) to the performance are enumerated and analyzed. Critical Control Points (CCP) are determined for these factors, such as depicted in Fig. 8. For each effective CCP, threshold(s) is set to provide the reference basis for monitoring and control. The mechanisms of proactive monitoring and control will then be in place. A system like this will feature information gathering and aggregation, context-awareness, learning and knowledge development, distributed policy-based control, consultation and decision making, correction and adaptation etc., depends on the level of autonomy/intelligence and the level of user interaction/intervention desired. Being a self-organized and self-govern system, verification and evaluation (e.g. fitness assessment) have to be conducted to ensure the correct functioning of the system. Naturally, the work along the chain will be noted, if so desired.

3.6 Determination of Critical Control Points

As distributed and policy-based monitoring and control is a most essential part of the model, and this hinges on accurately setting and effectively working critical control points, we explain the logical steps involved in the determination of a CCP in a more detailed way. As can be seen from Fig. 8, at the very beginning, a CCP is assumed. Should there be no control measure or if control is not any more necessary, the assumption of a CCP is dropped. Otherwise, if control is desired even if no measure being present, a control step is re-examined to find an appropriate measure. For each existing control measure, once it is confirmed that a control step can eliminate or at least reduce the possibility of the occurrence of a performance hazard, a CCP is established. Even if the current control step can not eliminate or reduce the likelihood of the occurrence of a performance hazard, and no subsequent control step is able to do so either, should excess of a metric will lead to a performance hazard, a CCP has to be established as well. Only if excess of a control threshold will not be a performance hazard, or there exists a capable subsequent control step down the path, will a CCP assumption be dropped in such context.

4 Conclusions and Future Work

We have reviewed the recent advances of protocol stack design in facing the challenges in managing complexity, emerging communication paradigms, and new performance requirements of the Mobile Internet. We pointed out that one promising direction to go is novel protocol stack design for Autonomic Communication. We then outlined our vision on how innovative organization of the communication software and the cross-layer optimized and context-aware protocol stack can help to realize such goal. The proposed POEM model places the

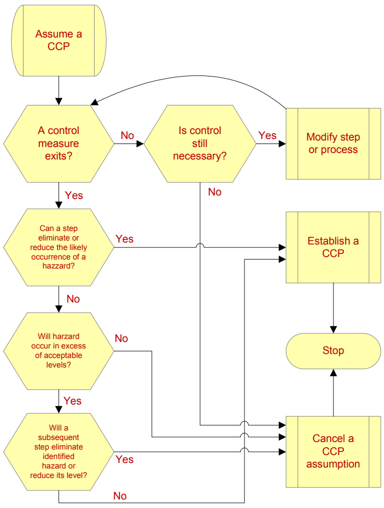

Fig. 8. Flow Chart on Critical Control Point Determination

cross-layer control functions beside the normal inter-layer interactions, so that ordinary features of the protocols are not compromised, but with the add-on benefits of well controlled optimization. Rooted from the observations of the nature ecosystem, we have applied some of the inspirations to the design of role-based composable functional system for self-optimizing AutoComm stack.

In addition to the metadata encapsulation and the entity-relationship we have coarsely described, a lot of issues are still open. We plan to perform in-depth investigation on the dynamic composition and decomposition of self-optimization CFS. Optimization data processing regulation and execution scheduling for conflict resolution and loop prevention is a must. Context awareness for effective communicating situation changes is definitely part of the design target, where Directed Diffusion, ACQUIRE and Reinforced Querying algorithms, linear and non-linear optimization methods might help. In distributed and proactive policy-based control, adaptive control theory and the principles distilled form natural ecosystems can be enlightening as well. As we have given only the procedure of CCP determination, other steps in the whole workflow of role-based CFS will be dealt with to complete the design.

The formal system modeling and specifications for POEM and simulation-based investigation of the performance gains are currently ongoing. We expect that the proposed reference model as well as the AC protocol stack design guidelines presented in this paper provide well-defined methodology at a critical time when new network technologies are on the cusps of mass proliferation. By doing so step-by-step, we envisage reaching the goal of self-tuning autonomic network with high level of autonomy and efficiency, with minimum human management complexity and user intervention.

Acknowledgment

This work has been partly supported by the European Union under the E-Next Project FP6-506869. In addition, we would like to thank OPNET Inc. for providing OPNET Modeler for the modeling of POEM.

References

1. R. Murch, *Autonomic Computing*. Upper Saddle River, New Jersey: Prentice Hall, Mar. 2004, iSBN:013144025X.
2. IBM, "Autonomic Computing Initiative," *IBM Press*, 2003, http://www.autonomic-computing.org.
3. J. Kephart and D. Chess, "The Vision of Autonomic Computing," *IEEE Computer*, vol. 36, no. 1, pp. 41–50, January 2003.
4. H. Schmeck, "Autonomic computing - vision and challenge for system design," in *Proceedings of the International Conference on Parallel Computing in Electrical Engineering (PARELEC'04)*, Dresden, Germany, September 2004, pp. 3–3.
5. C. Mller-Schloer, "Autonomic computing: on the feasibility of controlled emergence," in *Proceedings of the 2nd IEEE/ACM/IFIP international conference on hardware/software co-design and system synthesis (CODES+ISSS 2004)*, Stockholm, Sweden, September 2004, pp. 2–5.
6. "The Gryphon Project," http://www.research.ibm.com/gryphon.
7. "The SMART Project," http://www.research.ibm.com/autonomic/research/projects.html.
8. "The LEO Project," http://www.research.ibm.com/autonomic/research/projects.html.
9. "The Recovery Oriented Computing (ROC) Project," http://www.roc.cs.berkley.edu.
10. "The Software Rejuvenation Project," http://www.software-rejuvenation.com.
11. Cisco and IBM, "Adaptive Services Framework (ASF) White Paper," http://www.cisco.com/application/pdf/en/us/guest/partners/partners/c644/ccmigration_09186a0080202dc7.pdf.
12. "Autonomic Communication Forum," http://www.autonomic-communication-forum.org.
13. "Autonomic Communication Initiative," http://www.autonomic-communication.org.
14. "The Anthill Project," http://www.cs.unibo.it/projects/anthill.
15. I. Marshall, "UCL Bio-inspired Approaches to autonomous configuration of distributed systems," http://www-dse.doc.ic.ac.uk/projects.html.
16. C. Tschudin and L. Yamamoto, "A Metabolic Approach to Protocol Resilience," in *Proceedings of the 1st IFIP International Workshop on Autonomic Communication (WAC 2004)*, Berlin, Germany, October 2004, pp. 2–5.
17. M. Smirnov, "Autonomic Communication: Research Agenda for a New Communication Paradigm," *Fraunhofer FOKUS White Paper*, November 2004.
18. C. M., M. G., T. G., and G. S., "Cross-layering in Mobile Ad Hoc Network Design," *IEEE Computer Magazine*, vol. 37, no. 2, pp. 48–51, February 2004.
19. Z. J. Haas, "Design methodologies for adaptive and multimedia networks," *IEEE Communications Magazine*, vol. 39, no. 11, pp. 106–107, November 2001.

20. S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, "Cross-layer Design for Wireless Networks," *IEEE Communications Magazine*, vol. 41, no. 10, pp. 74–80, October 2003.
21. V. Kawadia and P. R. Kumar, "A cautionary perspective on cross layer design," *IEEE Wireless Communication Magazine*, vol. 12, no. 1, pp. 3–11, February 2005.
22. R. Braden, T. Faber, and M. Handley, "From Protocol Stack to Protocol Heap - Role-Based Architecture," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 17–22, January 2003.
23. Wang, Y and Touch, J. and Silvester J., "A Unified Model for End Point Resolution and Domain Conversion for Multi-Hop, Multi-Layer Communication," ISI, Tech. Rep. ISI-TR-590, 2004, http://www.isi.edu/~yushunwa/papers/isi-tr-2004-590.pdf.
24. V. Cerf, "The catenet model for internetworking," *Internet Experiment Notes IEN48*, July 1978.
25. J. Wroclawski, "The Metanet, Research Challenges for the Next Generation Internet," in *Proceedings of Workshop on Research Directions for the Next Generation Internet*, May 1997.
26. K. Sollins, "Designing for Scale and Differentiation," in *Proceedings of the Workshop on Future Directions in Network Architecture(FNDA) at ACM SIGCOMM*, Karlsruhe, Germany, August 2003, pp. 267–276.
27. J. Crowcroft, S. Hand, R. Mortier, T. Roscoe, and A. Warfield, "Plutarch: An Argument for Network Pluralism," in *Proceedings of the Workshop on Future Directions in Network Architecture (FNDA) at ACM SIGCOMM*, Karlsruhe, Germany, August 2003.
28. M. Borella, J. Lo, D. Grabelsky, , and G. Montenegro, "Realm Specific IP: Framework," RFC 3102, IETF, Oct. 2001. [Online]. Available: http://www.ietf.org/rfc/rfc3102.txt
29. Z. Turanyi, A. Valko, and A. Campbell, "4+4: An architecture for evolving the internet address space back toward transparency," *ACM Computer Communication Review*, vol. 33, no. 5, pp. 43–54, October 2003.
30. B. Tung and L. Kleinrock, "Using finite state automata to produce self-optimization and self-control," *IEEE Transaction on Parallel and Distributed Systems*, vol. 7, no. 4, pp. 439–448, 1996.
31. J. Gausemeier, "From Mechatronics to Self-Optimization," in *Proceedings of International Congress on FEM Technology*, Lake Constance, Germany, October 2002.
32. N. Kandasamy, S. Abdelwahed, and J. P. Hayes, "Self-Optimization in Computer Systems via Online Control: Application to Power Management," in *Proceedings of International Conference on Autonomic Computing (ICAC'04)*, New York, USA, May 2004, pp. 54–61.
33. B. Krishnamachari, "Self Optimization in Wireless Sensor Networks," in *Invited talk at the NSF-RPI Workshop on Pervasive Computing and Networking*, Troy, NY, April 2004.
34. S. Aiber, D. Gilat, A. Landau, N. Razinkov, A. Sela, and S. Wasserkrug, "Autonomic Self-Optimization According to Business Objectives," in *Proceedings of International Conference on Autonomic Computing (ICAC'04)*, New York City, USA, May 2004, pp. 206–213.
35. K. Farkas, O. Wellnitz, M. Dick, X. Gu, M. Busse, W. Effelsberg, Y. Rebahi, D. Sisalem, D. Grigoras, K. Stefanidis, , and D. N. Serpanos, "Real-time Service Provisioning for Mobile and Wireless Networks," Submitted to Elsvier Computer Communication Journal, Oct. 2004.