Enabling Communication Service 
Reconfigurability via Guided 
Cross Layering

Muslim Elkotob, Evgeny Osipov
Enabling Communication Service
Reconfigurability via Guided Cross Layering

Muslim Elkotob, Evgeny Osipov

Luleå University of Technology
Department of Computer Science and Electrical Engineering
Division of Systems and Interaction
Enabling Communication Service
Reconfigurability via Guided Cross Layering

M. Elkotob and E. Osipov
Luleå University of Technology, Luleå, Sweden
<Firstname>.<Lastname>@ltu.se

Abstract—A system architecture which natively supports cross-layer design in a general sense is an essential prerequisite for enabling communication services in the future heterogeneous Internet. A multitude of cross-layer approaches ranging from clean slate designs to actual implementations of cross-layer links in the standard TCP/IP stack have been suggested during the last decade. Yet, there is no agreement on a systematic integration of cross-layering into the actual Internet architecture. In this article we present a Guided Cross-layering Framework and a roadmap for its deployment in the Internet. We elaborate its key issue of identifying functional invariants in the present communication model. The invariants are the clean protocol stubs of the current TCP/IP stack. We describe the concept of meta-protocols and a design-deployment methodology for the framework. Our main postulate that promises an acceptance of the proposed architecture is the evolutionary, market-driven transformation of the current Internet architecture. On the example of the new ICSP (Integrated Communication and Signaling Protocol) protocol, we demonstrate the integration of CARD (Candidate Access Router Discovery), MIP (Mobile IP), and SIP (Session Initiation Protocol) in our framework which jointly optimize the performance and economic utilities of a multi-cell wireless network operator and the end user.

I. INTRODUCTION

A discussion on the next generation system architecture for communication networks is currently a very hot topic in the research community [10][18]. It is fair to say that the debates on the architecture for the Future Internet began more or less immediately after the TCP/IP technology became popular and the Internet services gained their momentum. Originally, the discussion was conducted mainly around the shortage of IPv4 addresses problem. Another turn of debates started with the appearance and enormously fast spread of new communication media such as WiFi, UMTS, and WiMax. Suddenly, the well established and highly optimized TCP/IP protocol stack started to deliver very poor performances when running over new media. It is soon became understood that the performance problems stem from highly dynamic characteristics of the radio channel. Solutions to these problems appeared to be problematic to implement in the scope of existing layered communication reference model. On the other hand, doing unsystematic media-specific fixes to the standard protocol implementations raise serious problems with backward compatibility and at the end make the overall system management and analysis intractable.

The two fundamental requirements for the future architecture are support for heterogeneous transmission media and its autonomic functionality. When addressing the first requirement the major challenge to resolve is how to organize a systematic cooperation between communication layers that is currently impossible due to strictly defined APIs of the present TCP/IP stack. The second requirement is a multisided problem that comes side by side with the first one. Amongst the major principles of autonomic communications we firstly highlight on-demand creation of new communication services in a general sense. These services include as new application layer services as well as the services provided by particular lower layer communication protocols. Secondly, in order to cope with heterogeneity and the overall complexity of the future Internet architecture self-* features must be natively supported by the design.

There are several proposals addressing the fundamental challenges ranging from actual implementation of particular fixes to the current protocols e.g. [25] to revolutionary clean slate design approaches e.g. [4]. In this article we present our vision on how the above stated design objectives can be gradually deployed in the current Internet architecture and allow for its evolutionary transition to a fully autonomous architecture of the future.

We present a Guided Cross-layering Framework and a roadmap for its deployment in the Internet. The framework is built around our key postulate that there exist functional invariants in the current Internet architecture. In technical terms, the invariants are the clean protocol stubs of the present TCP/IP stack.
We describe the concept of meta-protocols and a design-deployment methodology for the framework. This methodology relies on fundamental principles and accounts for rich experience of developing and deploying dynamic distributed software gained from the areas Service Oriented Architectures (SOA) and middleware. The guided part in the framework’s name reflects that the cross-layer optimizations do not happen spontaneously but are initiated during certain interactions between different players on the telecommunication market engaged in a particular relationship.

A. Contribution and structure of the article

Our main contribution is suggesting functional invariants as one of fundamental architectural abstractions and placing them in the context of the Internet eco-system. The major property that promises an acceptance of the proposed architecture is the evolutionary, market-driven transformation of the current Internet architecture.

On example of the new Integrated Communication and Signaling Protocol (ICSP) we demonstrate how a new compound service implemented on the intersection of session, network and link layers can be expressed within and be deployed using our framework. We show how technical performance objectives of several players on the communication market meet their specific economical utilities. The ICSP protocol represents our second contribution.

The technical report is organized as follows. Section 2 presents a detailed motivation for developing the guided cross-layering framework and overviews its functionality. Section 3 is the main section where we elaborate the concept of functional invariants and describe the concept of meta-protocols. Section 4 presents the ICSP protocol as an example of the framework application. We show how our proposal fits in the scope of existing related work and discuss the technological and scientific challenges that need to be addressed in the scope of our framework in Section 5 before concluding the article in Section 6.

II. MOTIVATION AND THE PROPOSAL OUTLINE

Parameter fusion: In essence, the majority of current works on cross-layer performance optimization concerns the run-time fusion of parameters across communication layers in systems with the traditional TCP/IP layered architecture. When operating over a wireless channel, the knowledge of the run time parameters from the MAC and the link layers is essential to boost the performance of the upper layer protocols, e.g. routing using link layer feedback. The awareness of the link conditions directly at the transport layer is essential in order to cope with the extremely poor TCP performance in the multi-hop wireless environment [23]. There are two major problems associated with this type of cross layering at the present time. Firstly, the identification of parameters that need to be communicated across layers is currently completely developer dependent. As advocated in [12], situations will lead to conflicts of internal protocol states when accessed by several concurrent cross-layer mechanisms. Connected to the first problem, the second problem concerns the implementation of such proposals. It is fair to say that all current implementations directly intervene with the standard protocol implementations and violate the basic principles of complex software engineering. As the simplest consequence this leads to difficulties to analyze the proposed modifications: It is virtually impossible to decide whether the performance gain or reduction is due to a new functionality or just a feature of an improper implementation. Most importantly, adding the developer specific interfaces to standard implementations unarguably leads to spaghetti system design and in a long run to complete system malfunctioning.

Compound protocols: Compound protocols represent another thread in the cross-layering paradigm. Functionality of several self-consistent existing protocols on different communication layers can be combined to achieve a richer functionality. Compound protocols appear mainly as a vendor-specific implementation phenomenon. For example implementations of multi-homing using mobile IP require continuous monitoring of the link qualities on available interfaces. This involves acquiring SNR values from the link layer and performing additional logic beyond the scope for Mobile IP specification. While it is relatively easy to integrate certain protocols using existing APIs, other protocols are hidden by strict layer APIs.

System architecture: It is currently well understood that the conventional layered system design paradigm that exists unmodified since the Internet had emerged places hard practical boundaries on passing such information across the layers. Currently there is a consensus in the networking research
community that specifying a “universal” static stack of network protocols for heterogeneous transmission environments is a very difficult if at all possible task. Instead, numerous research projects (e.g.[4][10][18]) are defined towards developing flexible system architectures in which the functional content of the communication stack is configured depending on the communication context of a particular user, e.g. type of communication environment, user's quality of experience for a particular data session in terms of delay, jitter, etc. The dream goal of such developments is self-configuring autonomous communication architecture. In this case, choosing a correct set of communication protocols is subject to cross-layer optimization on a system-architectural level. This concept is supported by recent formal frameworks for mathematical analysis of configurable protocols [6][7].

Invariants: While the idea of fully autonomic communication systems is attractive one has to keep as close to the reality as possible. The reality is that the Internet architecture is accepted and deployed worldwide. A multi-billion-dollar market is behind the current implementation of Internet protocols. We have to face the fact that the key stakeholders such as vendors of hardware and operating systems as well as the major network operators protect the Internet standards from drastic changes. The current Internet design has therefore a set of invariants that need to be accounted for when thinking about the Future Internet architecture. Invariants of a design as discussed in [1] are specific characteristics that limit its changeability on the one hand and as the authors argue, when identified correctly the invariants can provide a set of building blocks on which new capabilities and concepts can be build without requiring fundamental system restructuring on the other.

A. Motivation

In the light of the above reasoning we believe that revolutionary clean slate approaches towards new system architecture are less likely to happen in the foreseeable future. We share the view point of the authors in [1] and propose a framework for guided cross-layering by identifying functional invariants of existing Internet protocols. The key design objective of our framework is market orientation; we suggest incentives for major players on the communication market towards its gradual acceptance. In this technical report, we aim at presenting an alternative to a clean-slate design roadmap towards the creation of a flexible network architecture of the future. We emphasize that we do not suggest specific technologies as ultimate implementation solution rather we intend to raise a discussion on identifying the functional invariants and how to fill the framework building blocks with the particular content.

B. Proposal outline

The guided cross-layering framework, referred to as GXF further on, consists of a set of abstractions, concepts and a methodology for integrating functionality of existing network protocols for providing communication services on different communication layers. Figure 1 illustrates GXF’s main building blocks and the indices of the corresponding sections describing their details. The nucleus of the framework is a meta-protocol abstraction for specification of compound communication services on different layers of the OSI reference model. One example of a compound service is TCP with a particular flavor of congestion control that relies on current characteristics of transmission channel. Another example is a protocol that bases its functionality on collaborative work of MobileIP and SIP protocols to reduce the probability of session interruptions during network layer handovers. The second
component of the GXF framework is a semantic consistency and functional correctness verification toolbox. This component is responsible for pre-deployment verification of the compound service and also decides on the physical capability of the host operating system to deploy the offered service. If the output of this step is positive the operating system automatically performs dynamic wiring of the functional blocks suggested in the formal description to obtain actual OS-specific implementation. The blocks and the transition procedures marked by dashed lines jointly represent the performance optimization feedback loop. We include the runtime-testing as one of the primary objectives for the GXF framework in order to enable the autonomic self-* features.

The GXF framework is intended for usage by different players of the communication market, namely, service providers, access network operators, vendors of operating systems, end-users, developers of network services in a general sense to automatically (re-)configure the content of the communication stack in order to meet their specific utility objectives. We discuss the techno-economical relationships between communication market stakeholders and actors as well as give examples of GXF in Sections 4 and 5.

III. THE GUIDED CROSS-LAYERING FRAMEWORK: CONCEPTS AND METHODOLOGY

A. Functional invariants as prerequisites for GXF

Functional invariants exist in the design of complex systems in all areas of current technology. As a very simple intuitive example, take bicycle design. As a matter of fact, their basic shape and configuration has changed little since the first chain-driven model was developed around 1885. Add a new gear system, disc brakes or use carbon material – the frame, two wheels and the chain remain invariant. The new features of the variant parts simply increase the bicycle’s performance characteristics such as speed, weight, and security. Without going into broad philosophical discussion, functional invariants exist also in the current design of the Internet architecture. The challenge is to correctly identify them. We stress the fact that it is not our ambition to uniquely identify all functional invariants of the Internet architecture in this article. Rather, we intend to demonstrate some key invariants of selected network protocols and put them in a systematic framework to add variant parts in order to enrich the functionality of the networks and increase the performance of communications.

One of the most illustrative protocols where the functional invariants could be demonstrated is TCP. There exist several versions of the protocol officially accepted for implementation by different OS vendors (e.g. TCP NewReno [27], TCP Westwood [26]). Since the time when wireless technology emerged and new research fields such as mobile ad-hoc and mesh networks appeared, the number of distinct implementations adapting the standard TCP to the specifics of radio channel has exceeded several dozens. In fact, all accepted as well as new experimental TCP versions (even those that heavily rely on cross-layer interactions) could be classified into the finite set of categories [23] and the references therein: Modification to congestion notification procedures (e.g. delayed or adapted acknowledgements); Modifications to the Slow Start phase; Modifications to the congestion control procedure. What remains invariant in the case of the TCP protocol are its connection control state machine and the presence of states that invoke Flow Control, Slow Start, Congestion Control phases.

**Definition 1.** Key states of a protocol are the internal states that determine the border between the invariant and variant parts of the protocol’s Finite State Machine (FSM). We postulate that all other internal states of a protocol are irrelevant in the scope its evolution.

**Definition 2.** Functional invariant of a network protocol is a finite state machine of invariant and key states where the state machines describing the functionality of the key states are decoupled from the protocol’s FSM.

In other words, functional invariants are the stubs of the protocols in the present TCP/IP stack where key states are exposed in their corresponding application programming interfaces.
B. Meta-protocols as formal specification of communication services

Figure 2 illustrates the GXF definitions given below. Our guided cross-layering framework is built around a notion of a communication service. While there are several definitions of this term, here we understand it similarly to [16]:

**Definition 3.** Communication service is a simple or composite functionality provided by one logical entity to another.

An example communication service would be a functionality of the particular network protocol residing on one specific layer of the TCP/IP stack like IP, TCP, etc.; or a service provided by an interplay of several communication protocols like DHCP involving IP and ICMP. The complexity of a communication service is described by its communication scope.

**Definition 4.** Communication scope of a communication service is a non-zero set of communication layers as defined by the seven layers OSI reference model containing functional invariants involved in the delivery of this network service.

We advocate that despite of the wide popularity of the TCP/IP communication model accepted for actual implementation OSI model gives richer semantics when describing existing communication protocols in the Internet. It is important to note that functional invariants included in a specific communication service may in turn be synthesized with other invariants, these second tier invariants are seen as default configuration of the communication service. For example, when integrating SIP and mobile IP to provide a new integrated service we may rely on default configuration of other lower-layer protocols and services. An example of the communication scope would be SCOPETCP=<L4, L2>, when thinking of link-layer dependent congestion control modifications. When discussing the GXF usage philosophy in Section 5, we note that the scope is also restricted for services offered by different players on the telecommunication market.

**Definition 5.** Meta-protocol is a finite state machine describing a communication service in a particular scope and the associated sequence of meta-PDUs triggering state transitions. A finite state machine of a meta-protocol is a state machine of the communication service logic concatenating the state machines of particular communication protocols included in the scope. The concatenation of state machines is done through key-states of state machines of involved protocols.

**Definition 6.** Meta-PDUs are protocol data units of actual communication protocols included in the communication scope of the communication service.

C. A toolbox for specification and verification of meta-protocols

There are three major phases of a communication service: Its formal specification, implementation and deployment. The first phase requires a formal description language. Formal verification of specification’s correctness should precede the transition to the second phase. Finally the consistency of the offered implemented service should be verified by the target host before actual deployment.

---

1 The terminology used in this technical report such as meta-protocol, meta-states, and meta-stack has barely been used in the same context as we do. The authors in [11] published a decade ago, describe a meta-protocol as a mean to aggregate inputs from several MAC protocols and generates a common notation.

2 In the present Internet there are many protocols which functionality falls beyond the semantics of the TCP/IP model. For example Session Initiation Protocol is more suitable to characterize as Session Layer protocol in the OSI model than Application layer protocol in the TCP/IP reference. Another example would be good to have here.
Currently network protocols are described in a free textual form in Request for Comments (RFC) documents. Some (but not all) protocols such as TCP have also well-specified Finite State Machines. Obviously, this is not enough to systematically aggregate several protocols into a compound service and formally verify its correctness.

We propose to use rich experience collected in the domains of Service Oriented Architectures and middleware design [19][24] to create a unified toolbox for specification and verification of meta-protocols during the life cycle of a communication service. There are several existing approaches towards formal specification of and reasoning on complex systems in the area of Software Engineering. In this article we do not argue in favor of one or another methodology. As an illustration of a technique that can be deployed in our framework we choose tree based ontologies to systematically represent the set of functional invariants. Figure 3 presents a simplified ontology of network protocols involved in the example of GXF usage in Section 4. This nomenclatural taxonomy of protocols for a particular category or functional area (in our case Mobility and Motion) sorts protocols and determines the interdependencies between them when a service is created. Each node has a set of key states and their corresponding APIs as leaf nodes for a particular protocol. Those states and APIs shape the behavior of the compound meta-protocol.

The formal semantic description of functional invariants (in our case ontologies), the associated reasoning-verification methodology and formally defined finite state machines of the functional invariants along with the API to their key-states assist the service developer to correctly implement the service. Once implemented the service should be passed to the target consumer (e.g. the end user when the service concerns modifications to the functionality of the end host). Here the service should be verified for functional consistency in the scope of consumer’s operating environment. For this purpose a formal specification of the ready to deploy service is needed. The same semantic description is now used when analyzing the service specification for compatibility verification. We describe an example of navigation through the ontologies for creating new communication service, specification of its implemented version and the deployment procedure at the consumer side in the next section in details.

IV. DEMONSTRATION USE CASE: ROAMING USER IN A BROADBAND WIRELESS NETWORK

We decided to demonstrate the usage of the GXF framework by describing the creation and the runtime deployment of a new meta-protocol. Consider mobile users entering an access network of a
virtual operator covering a broad region and consisting of several WiFi based cells. The user is engaged in several multimedia real-time sessions on his mobile terminal and traverses several adjacent cells. On the user’s terminal the SIP protocol takes care of multimedia session management and session mobility and MobileIP (MIP) implements the network-layer mobility. The Candidate Access Router Discovery (CARD) [15] protocol running both on the client and network sides collects the information about neighboring access points (that also act as routers) in the coverage range of a particular access point. The mobile terminal and the access points are aware of their geographical position using GPS. The network’s role in this scenario is minimized to simply provide connectivity.

A. Motivation: When performance and economical objectives of different actors meet

While a general discussion on the relationship between different communication market players and their connection to the proposed GXF framework is presented in Section 5, here we give a first illustrative example which motivates two actors being involved in our framework. Without any modifications when the protocols work in an uncoordinated way MIP will enter the handover state as soon as the current link will go down or the terminal will be connected to the network of the new cell which is chosen with the help of the CARD protocol. In its turn SIP will independently discover the new location via a SIP proxy server and enter the re-invite state. As it is easy to imagine, the potential problem of these uncoordinated actions is large handover times and session interruption in the worst case. Even when the higher level handover is successful and is within the acceptable boundaries the next (random) cell might be overloaded and the session will interrupt anyway. One straightforward way to improve the situation in this case is to stay connected to several overlapping cell, this however would increase the cost of communications due to over-provisioning on the client side. Since the network role is minimal such solutions for micro and macro mobility as Cellular IP and HAWAII [13] are not an option in this scenario.

On the other hand, the obvious objective of the mobile user is to preserve the quality of the connection and not overprovision as much as possible. As for the operator the objective is also to avoid bandwidth waste due to client over provisioning and at the same time keep the network infrastructure as simple (and therefore cheap and easy to manage) as possible. If upon entering the operator’s network the functionality of the mobile terminal would be updated to efficiently roam in this particular network with minimal changes to the client’s operating system, both parties will win in corresponding performance and subsequent economic benefit.

B. ICSP description

The Integrated Communication and Signaling Protocol (ICSP) is reflected more formally in the high-level pseudo-code of ICSP below:

```
//ICSP Service Start Phase
1  connect_to_network (MIP)
2  launch VoIP (SIP) //multimedia session with SIP signaling
//Service during mobility
3  if link_quality_ok
4      return;
5  if link_going_down {
6      get_motion_info [get velocity, get direction, get coverage] (GPS)
7      get_candidate_network (CARD) //using motion information
8      set_candidate_network (MIP) //using CARD info
9      mobility_binding (MIP) // MIP RFC 2002, address association
10     re-invite (SIP) // Multimedia session update on location and settings
11     return //go to step 3
// ICSP Service termination
12 SIP_BYE // close multimedia session
13 network_disconnect //
```

Note that as protocols are wired in our proposed methodology, identifying the relevant states is a key issue at this point. An example would be when the GPS module gets location information and then CARD uses this information in order to know which neighbors to check for their availability for a handover. This selection process of candidate network is called get_candidate_network and is reflected in the pseudo-code as well as in the ontology in Figure 3. In turn the MIP protocol that needs to know to which network to select as a handover gets this information from CARD after wiring and
this is reflected in the set_candidate_network in the same figure and pseudo-code. Our methodology is used to wire all relevant protocols to yield the desired system behavior as specified in the service logic and agreed upon between the network operator, service provider and the user.

C. Using the GXF framework in the roaming user scenario

We first describe the development phase of the ICSP service. The development using a service composition tool identifies the relevant to the service ontologies. In the case of the ICSP meta-protocol these ontologies are for categories “Mobility” and “Motion” (see Figure 3). The “Mobility” ontology identifies Mobile IP, H.323, SIP and CARD protocols. The “Motion” ontology identifies GPS and GPS_Extension_X services. She finds that Mobile IP has handover preparation and re-connected as key states and register, update, and get candidate network as the standardized APIs to access the key states. As the key states and the corresponding APIs for other services are indicated in the figure, we do not describe them here.

The service logic depicted on the left side of Figure 3 describes the functional domains to which the relevant invariants belong. These categories are the root nodes of our ontologies. The figure also illustrates the mapping of the identified key states of selected protocols as the result of navigation through the relevant ontologies. The developer implements the ICSP service for several target operating systems as pluggable modules and describes the implementation as shown in Figure 4. The OS specific binary modules of the meta-protocol as well as the service description are kept in the operator’s repository.

The deployment phase starts when a mobile user enters the first cell of the virtual operator. The operator offers the certified service description to the user. The “Description” part contains the textual description of ICSP functionality and the techno-economical incentives for the target consumer. The user controls the authenticity and trustworthiness of the service. It checks also the scope of the proposed meta-protocol. As we discuss further in Section 5, the scope of meta-protocols should be restricted for different actors. The “Features” part of the description gives an indication to the roots of particular ontologies kept in the terminal for fast navigation. These terminal ontologies and the navigation procedure are the same as for the developer as described above. The terminal goal, however, is to identify the actual internal modules that have to be wired in order to deploy the service. The
“Implementation” part contains the actual location of the ICSP binaries for the specific operating system of the host. Finally, the “Wiring-information” part instructs the host how to wire the ICSP module to the binary modules of protocols installed on the user terminal. The wiring of the components is performed dynamically by the OS. The wiring related issues are further discussed in the next section. Note that the mobile terminal may deny installation of even a certified service offered by the third party due to several reasons. Firstly, the user may simply not want to do this. This indicates that the technoeconomical incentives are not attractive for her. Secondly, the mobile terminal during the navigation through the ontologies finds that it does not have the required functional invariants. Finally, the problems during wiring due to improper implementation for particular version of the user’s OS of the service may occur. In the latest case the terminal’s OS would return to the default stable configuration of the protocols. In either case the user will still be able to roam through the operator’s network this time, however, in a less efficient manner as described in the beginning of this section.

V. DISCUSSION AND RELATED WORK

A. Identification of functional invariants

Identification of functional invariants is a challenging task. We identify the way towards determining the functional invariants of the current protocol stack through their concise definitions. A good example is a description of TCP Westwood from the website of the protocol [Westwood]: “TCPW relies on **mining the ACK stream** for information to help it better set the congestion control parameters: Slow Start Threshold (ssthresh), and Congestion Window (cwin). In TCPW, an "Eligible Rate" is estimated and used by the sender to update ssthresh and cwin upon loss indication, or during its "Agile Probing" phase, a proposed **modification to the well-known Slow Start phase**...”. This definition clearly identifies the Slow Start and the Congestion Control phases as variant parts and the congestion indication (ACK stream) as one of the key-states for TCP. When analyzing the existing experimental TCP versions including those which heavily rely on the cross-layer interaction, it is feasible to spot common modification patterns and the functional parts of the protocol which remain invariant. We believe that this type of analysis of major protocols of the TCP/IP stack would reveal a finite set of functional invariants.

B. GXF usage philosophy and the associated challenges

The essential component of the GXF framework is the performance-evaluation feedback loop as shown in Figure 1. This part of the framework is used in the development and post-deployment phases of the communication services’ life cycle. In the development phase the loop is intended for formal analysis cross-layer meta-protocol performance. We particularly refer to the mathematical theory of cross layer optimization in [6]. There the authors present a framework for analysis of communication networks by decomposing each layer into a set of utility maximization sub-problems. This theory is essential for identifying the inter-dependencies between parameters of variant protocol parts on different communication layers.

As for the post-deployment phase within the GXF framework we envision a set of standardized performance tests for all existing meta-protocols running on the target system. These tests will evaluate the run-time quality characteristics of protocols’ variant parts in different communication scenarios. The variants will be ranked according to the actor’s quality of experience metrics. The collected information is then fed into a publicly accessible repository for future use by other systems that find themselves in a similar communication context. Self-reconfigurability is on the key principles of autonomic communications. Without diminishing the important role of simulations in the performance evaluation of network protocols, we also emphasize that this operation mode of the GXF framework is capable to finally eliminate their dominating role as the ultimate framework for experimental network analysis.

The important aspect of the GXF framework is its “guided” nature. In essence, this means that particular cross-layer interactions do not happen spontaneously. Meta-protocols are relevant within business relationships between particular players of the communication market. In the previous section we demonstrated the creation of a communication service based on meta-protocols initiated by a network operator. Another example would be a construction of a meta-protocol by a provider of a particular multimedia content that suggests characteristics of the encoder depending on the quality of a specific communication session [9]. A vendor of an operating system would create lower layers meta-protocols, for instance supplying different variant parts of congestion control for TCP that depend on the type of communication medium. The important aspect in this context is to correctly identify the
communication scope for meta-protocol supplied to the consumer by a particular third party. While a user would likely allow the OS vendor to reconfigure protocols on all layers, she would rather prefer that operator or content provider meta-protocol do not touch the basic system functionality and operate only on selected upper layers of the stack.

Obviously, a particular meta-protocol might be incompatible with the actor’s current system configuration or the implementation of the service is simply erroneous. In order to ensure system robustness and the safe fall back when errors occur, in the GXF framework we require that the last stable and the default configurations of all system’s meta-protocols are kept locally on the target host. We also require that the default configuration of meta-protocols is equivalent to the current standard content of the TCP/IP stack. Since we identify the functional invariants for the current Internet protocols this requirement is automatically feasible within the GXF design. The latest requirement also insures backwards compatibility at least during the initial transition to the new Internet architecture (until one or several functional invariants will change) even with traditional systems completely that are completely outside the GXF framework.

C. GXF implementation aspects: Wiring of invariant and variant parts

So far we intentionally have not specified a particular way of GXF implementation. There are several proposals for implementing the cross-layer design. A good overview of the existing approaches is given in [25]. We, however, see a large potential for GXF implementation through dynamically linking modules analogously to DLL libraries in Microsoft Windows or kernel modules in Linux OS. While modularization of existing solid implementations of network protocols would certainly lead to an increased computational overhead, we advocate that it would hardly be a problem for modern high to medium end hardware. In this respect we see an important role of open source operating systems as being more reactive for implementing innovative solutions.

D. GXF in the scope of a market driven holistic eco-system

When talking about Future Internet Architecture business relationship within a particular proposal is the essential issue to address. While related to the GXF clean slate design proposals [4],[10],[18] have definite chances to find more elegant solutions due to abstracting away from the limitations of the current design, the economical incentives for the major players on the communication market are not very stringent. In the real world’s multi-billion communication market that heavily depends on the current implementation of network software the new architecture will be viable only when each stakeholder and player will see clear techno-economical benefits. While a comprehensive discussion on business issues of the telecommunication market goes much beyond the scope for this report, here we present a simplified holistic eco-system view to illustrate the market driven nature of the GXF framework. Figure 5 shows a user centric relationships between operating system vendors, system developers, Internet Service Providers (ISP), and network operators, and service and content providers.
that deal with the end-user service logic and presentation rather connectivity and network issues. While being heavily simplified, this model is not very far from the reality. Services are becoming more and more user-centric, as is witnessed in current trends and as has been stated as the key feature of the Fourth Generation Networks (4G).

In the model every stakeholder is driven by a technical utility (TU) and a market utility (MU). A technical utility refers to the functional parameters that reflect performance of the player’s system and which should be optimized according to certain technical objective. On the other hand, the market utility reflects the business or profit aspect of a particular stakeholder when involved in particular relationships with other actors on the market. The TUs and MUs relate to each other, in the sense that one drives the other and vice versa. For instance, for OS vendors, the TU would be high system performance, whereas the MU would be a good profit margin due to mass market deployment and adoption. At the same time for the user will technically benefit from the adaptable performances offered by self configurable network functionality of the operating system and economically by minor updating the existing operating system. In Section 4 on a particular example we demonstrated the techno economical benefits for network operator and the user. Other MU and TU examples are shown in the figure for the remaining stakeholders.

In our model we emphasize the decisive role of the standardization bodies. Standardizing the functional invariants and the essential components of the framework is essential for its viability. Being aware that the key stakeholders on the telecommunication market has a great influence on the standardization process we believe that the suggested and further developed techno-economical benefits model of the GXF framework promises enough incentives for acceptance of the functional invariants approach by the standardization authorities.

E. The GXF framework in the scope of related work

In [20], the authors provide an extensive overview of the design choices of the current Internet and outline the design goals for the Future Internet and the challenges they face. Among the major challenges are interoperability and accommodating several modules into one architecture or protocol stack. The authors emphasize that the ITU-T has several abstract standards for the design of the next generation networks, however, there has not been much seen beyond those guidelines. Standardization is also addressed by the paper as the key issue.

Our guided service is based on major principles of component and service oriented architectures [19], [24]. Among the recent developments in the area of SOA with application to network functionalities the work in [5] addresses the service provisioning issue in multi-hop cellular networks. The paper provides several alternatives for service architectures using the well-defined SIP-Servlets framework. Certain analogies could be established also between the GXF framework and middleware architectures. In [2], for example, the authors elaborate the issue of mutual network-service awareness in evolving 3GPP networks using smart middleware. Cross-layer design is also a central issue in the proposal. We emphasize that we do not oppose our framework to the COA, SOA or middleware technologies. Moreover, we conjecture that the design of even lower layer network protocols should be conducted using a unified service oriented methodology equally applicable to development of application layer services and the distributed software in general.

Recently, there are several related to the GXF framework appeared in the research community. One notable work in [16] proposes an idea of having standard meta-data for improving the interoperability of different systems, subsystems, and services. The authors advocate that cross-layer challenges are better resolved when using semantic reference structures widely adopted and standardized. Applied to sensor and embedded networks the authors suggest a pre-compile time design and evaluation methodology. We also share the argument of this work on the significance of having standardized meta-data description of network components and configurations.

Several clean slate approaches are currently under development in Europe and the USA [4],[10],[18]. The common theme in these initiatives is an architecture that enables autonomic system behavior including dynamic adaptation to user and application requirements. Our approach while sharing the design objectives suggests a roadmap for gradual upgrade of the current architecture. To the best of our knowledge the proposed GXF approach is an original attempt to place a general problem of cross-layer design and systematic development of compound communication services on different layers of the present communication stack in the framework of business relationships between players on the communication market.
VI. CONCLUSION AND OUTLOOK

In this article we presented our vision on future development of the Internet architecture. We focused on particular and at the same time the essential aspect of dynamic (re-)configuration of network protocols. In a form of the guided cross-layering framework we described an alternative to a clean slate design road map for evolutionary transformation of the current Internet architecture incorporating the principles of autonomic communications. We advocated that in order for the future architecture to be accepted and be viable it must: Be deployed gradually; Be backwards compatible to the current architecture at least during the initial phase; It should create economical incentives for major players on the telecommunication market; and Finally, be supported by a strong standardization effort. Identification of functional invariants is the main pillar of the proposed architecture, Functional invariants are the clean protocol stubs of the current TCP/IP stack. Using the suggested concept of meta-protocols and the design-deployment methodology we showed a flexible, stakeholder-initiated, system reconfiguration at run-time without requiring fundamental system restructuring. On the example of the new ICSP protocol, we demonstrated the proposed performance and market driven incentives essential for practical evolution of our cross-layering framework. Finally, we emphasized the technological and scientific challenges that must be resolved in the scope of the GXF framework. We conclude that these challenges can only be resolved by a cooperative effort of all involved stakeholders.

References

[18] National Science Foundation (NSF) Future Internet Design (FIND) initiative: http://www.nets-find.net/
[22] Open Mobile Alliance: www.oma.org