Report from Dagstuhl Seminar 13131

Future Internet

Edited by

Jon Crowcroft[1], Markus Fidler[2], Klara Nahrstedt[3], and Ralf Steinmetz[4]

1 **University of Cambridge, UK,** Jon.Crowcroft@cl.cam.ac.uk
2 **Leibniz Universität Hannover, DE,** markus.fidler@ikt.uni-hannover.de
3 **University of Illinois – Urbana, US,** klara@illinois.edu
4 **TU Darmstadt, DE,** ralf.steinmetz@kom.tu-darmstadt.de

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13131 "Future Internet". At the seminar, about 40 invited researchers from academia and industry discussed the promises, approaches, and open challenges of the Future Internet. This report gives a general overview of the presentations and outcomes of discussions of the seminar.

Seminar 24.–27. March, 2013 – www.dagstuhl.de/13131
1998 ACM Subject Classification C.2.1 Network Architecture and Design – Packet switching networks, C.2.6 Internetworking – Standards
Keywords and phrases Future Internet, Network Design, Testbed, Software Defined Networking, Virtualization, OpenFlow
Digital Object Identifier 10.4230/DagRep.3.3.75

1 Executive Summary

Jon Crowcroft
Markus Fidler
Klara Nahrstedt
Ralf Steinmetz

The recent vision of the "Future Internet" attracts significant networking research and causes controversial debates on the actions to be taken. Clean-slate initiatives envision a fresh start that put fundamental principles of networking into question. Avoiding any constraints of the current Internet implementation, the ambition of the clean-slate approach is to understand and design the 'right' network architecture. Evolutionary approaches, on the other hand, seek incremental improvements, assuming that the Internet can –as in the past– be fixed to accommodate the changing needs of users and applications.

Numerous initiatives on the Future Internet, like FIND, GENI funded by the NSF, FIRE, 4WARD by the EU, and G-LAB by the BMBF, reflect the importance of the topic. Characteristic for numerous Future Internet initiatives is an experimental approach using testbed facilities such as the GENI or the G-Lab platform.

Challenges that are of central importance for the Future Internet fall into the following categories:

Network design: computer networks and the Internet obey certain architectural guidelines that reflect experience gained in the art of network design, such as layered reference models or the Internet end-to-end argument. While these principles are backed up by the success of the Internet, it has to be noted that the network exhibits major architectural restrictions, e.g., regarding mobility, security, and quality of service. Computer networking as a relatively young field of research can benefit significantly from architectural reconsiderations that are initiated by clean-slate initiatives. While today, network theory is largely descriptive, this Dagstuhl seminar investigated the potentialities of a prescriptive network theory, which could justify a methodical rule/equation-based approach for the design of future networks.

Virtualization: the virtualization paradigm revolutionized the use of computers and data centers, where the flexibility and mobility of virtual machines offers tremendous potential, posing, however, significant new challenges for networking. On the other hand, the virtualization paradigm has already many applications in networking, e.g., in virtual private networks or overlay networks. Currently, virtualization finds its way into network components, e.g., routers, itself, where the decoupling of logical entities from the physical substrate enables major innovations, e.g., concurrent (possibly post-IP) networks, infrastructure as a service, redundant shadow configurations, in network management, and in energy efficiency. Furthermore, the provisioning of service-oriented virtual networks across multiple infrastructure providers creates the need for separation between the network operations and the physical infrastructure. This is expected to change the way that virtual networks are managed, debugged, and operated. The Dagstuhl seminar contrasted different approaches to network virtualization and investigated their applications.

Experimental research: the Internet standardization process relies on running code and real world verification. An essential prerequisite for the transfer of research results is building of large scale testbed networks. These are frequently implemented as virtual, Software Defined Networks that run concurrently to a production network using the same hardware. The Dagstuhl seminar revisited the experimental approach and gathered lessons learned and best practices.

During the seminar, we discussed and (partly) answered the following questions:

Is a prescriptive network theory feasible? Today, network research is largely descriptive, e.g., there exist methods and tools to model communication networks and protocols, to analyze their performance, or to verify their correctness. The design of new networks, however, lacks a prescriptive network theory that provides necessary rules and equations that specify how a network for a given purpose has to be built. Instead, network design relies heavily on previous experience and best practices frequently resulting in incremental works. In contrast, the clean-slate Future Internet approach seeks to build a new Internet architecture from scratch. In this case the design space is entirely open requiring decisions regarding functional and non-functional aspects, e.g.,

- Where to implement reliable/unreliable and connectionless/connection-oriented?
- Where (end systems or network) and in which layer to keep state information?
- Where and how to achieve security, quality of service, and dependability?
- How to split locators and identifiers?

Given the examples above, we discussed:

- How can a prescriptive approach to network theory be formulated?
- What are the perspectives and the fundamental limits of the candidate approaches?
- What are the prospects of the approach if successful?

Which insights can the experimental, testbed-based approach reveal? Many approaches to the Future Internet are experimentally driven and centered around a testbed that ideally if successful becomes the first running prototype of the Future Internet. Clearly, testbeds are indispensable to implement running code as a proof-of-concept, whereas their use for understanding networking and for establishing new principles and paradigms can be debated. In the seminar we elaborated on this question to provide answers to:
- Which insights can be expected?
- Which exemplary fundamental insights did emerge from testbeds?
- For which use cases are testbeds meaningful, e.g., to engineer details, to approach concepts weakly understood, to understand the impact of users, etc.?
- How should a testbed platform look like, which properties must be provided?
- How can testbeds be benchmarked to achieve comparability and validity?

What are the challenges for wide-area service-oriented virtual networks? The virtualization paradigm gained a lot of attention in networking as it provides numerous useful applications and promises to solve a number of important issues, such as the gradual deployment of new networking solutions in parallel to a running production network. Considering existing networking technologies, it becomes apparent that virtual networks and virtual network components are already being used in a multitude of different ways and in different layers, e.g., Virtual LANs (VLANs), Virtual Private Networks (VPNs), the Virtual Router Redundancy Protocol (VRRP), or in form of overlay networks to name a few. Furthermore, the abundance of resources offered by commodity hardware can turn it into a powerful and highly programmable platform for packet processing and forwarding. The virtualization of such programmable network elements can provide network slices which are highly customized for particular network services and applications. The topics that were discussed at the seminar include:
- Resource discovery and provisioning of virtual networks across multiple domains, given that infrastructure providers will not be willing to expose their topology, resource information and peering relationships to third-parties;
- Virtualization of network components (e.g., resource allocation, isolation issues);
- Scaling of virtual resources to meet variable service demand;
- Use cases of network virtualization.

This report provides an overview of the talks that were given during the seminar. Also, the seminar comprised a one minute madness session for introduction and for statements on the Future Internet, a breakout session for group work on the topic of prescriptive network theory, as well as podium discussions on experimentally driven research and on the use cases of SDN. The discussions, viewpoints, and results that were obtained are also summarized in the sequel.

We would like to thank all presenters, scribes, and participants for their contributions and lively discussions. Particular thanks go to the team of Schloss Dagstuhl for their excellent organization and support. We also would like to thank Anil Madhavapeddy for his feedback and comments on SDN.

2 Table of Contents

3 Overview of Talks

3.1 Prescriptive Network Theories

Jon Crowcroft (University of Cambridge, UK)

We have many descriptive theories which help us with the design process for future networks, but mostly they work as descriptive or analytic tools rather than generative systems.

What we want is some way to feed in some constraints and out comes a prescription for a network architecture – we can then fire up the traditional tools (graph theory, queueing theory, control theory, etc) on the various components, to get a more detailed understanding of how the system will operate.

Key ideas that might go into such a theory come from the work by John Doyle on Highly Optimised Tolerant systems, and in particular, a recent idea on "Constraints that de-constrain", such as the choice of the lowest common denominator in the layered architecture of the Internet as the waist of the hour glass. And the use of unstructured addresses in the original choice of how IP destinations (and sources) might be interpreted.

How could we embed such ideas in a generative theory?

3.2 Introduction to Prescriptive Network Theory

Markus Fidler (Leibniz Universität Hannover, DE)

The term 'prescriptive network theory' is not commonly used and it may be the lack of such a theory that explains fundamental uncertainties in network design. A prescriptive theory, as opposed to a descriptive theory, is concerned with what should be, rather than with what is. Prescriptive theories are often viewed as high-level guidelines or rules, e.g., for the design of systems. Formally, a prescriptive theory is axiomatic. As such, it is a framework to derive certain conclusions, yet, it cannot be proven in itself. This presentation gives an introduction to the topic and raises a number of fundamental questions for group work discussion.

3.3 Software-Defined Networking: Challenges and Opportunities

David Hausheer (TU Darmstadt, DE)

Main reference D. Hausheer, A. Parekh, J.C. Walrand, G. Schwartz, "Towards a compelling new Internet platform," Integrated Network Management 2011:1224–1227.
URL http://dx.doi.org/10.1109/INM.2011.5990569

One of the biggest challenges related to the Future Internet in general and Software-Defined Networking (SDN) more specifically, is that the current Internet architecture is ossified, i.e. it can hardly be changed. New protocols to improve security, QoS, or mobility rarely get deployed across administrative boundaries. Moreover, fundamental problems of coordination among providers lead to implementation within a single domain only. Such issues include interoperability of network elements, resolve conflicts of interest, and revenue sharing. SDN as a new technology will face the same challenges, i.e. a major problem will be to support deployment of SDN across multiple domains. This requires amongst other issues a "East/West"-SDN-Controller Interface to address coordination problems among multiple providers.

3.4 <provoke>Research Testbeds Considered Harmful</provoke>

Markus Hofmann (Alcatel-Lucent Bell Labs – Holmdel, US)

The best research solves the best problems. To find the best problems, look within the known wide-spread or large-scale problems. Almost everyone will know about the high-level problem; it is the successful researcher who looks at the details to find a game-changing specific problem whose solution leads to a breakthrough.

In our profession, many of the best problems become apparent only when a system has been deployed for large-scale use. For example, it is hard to predict the creativity of hackers in a lab behind closed doors. Scalability issues often become apparent only after millions or billions of users adopt a solution and start relying on it for their daily life. Cost effectiveness – often one of the hardest problems, often leading to exciting inventions – raises in importance when commercial deployments are being considered. Resiliency of a network is non-trivial to determine on a whiteboard or even in a lab. How do the various components that make up a network react to dirt, dust, extreme temperatures? Will they hold up? Will we have to design the system differently? All these questions have to be addressed when designing networks for the real-world. They are some of the hardest problems to solve.

As such, experimental research, running code, and large-scale tests are key ingredients to making progress, having real impact, being relevant. While lab experiments provide initial insights into the feasibility of an approach, they rarely can address questions around scalability and resiliency. As a result, we often set-up larger-scale overlay testbeds that run as kind of a virtual network on top of production networks.

Using examples from earlier work with experimental overlay testbeds and real-life production networks, this talk will address the need for experimental research but also discuss the challenges and shortcomings of overlay-based approaches.

3.5 Wide-Area Distributed System Deployments Yield Fundamental Insights

Brad Karp (University College London, UK)

The first decade of the new millennium saw a burst of concentrated research activity on wide-area distributed systems: the handful of core Distributed Hash Table (DHT) designs; a rich set of applications built atop DHTs; peer-to-peer file sharing schemes such as BitTorrent; distributed network measurement services such as iPlane and Hubble; and content distribution and caching systems like Coral, CoDeeN, and CoBlitz. While it was clearly "fashionable" to do research on these topics during this period (especially in the DHT and P2P area), I believe another factor contributed to this explosion in activity: the widespread availability of wide-area testbeds like PlanetLab and RON.

In this talk, I will briefly review my experience (particularly with the OpenDHT DHT service, which ran on PlanetLab and was publicly accessible between 2004 and 2009) and that of others deploying systems on wide-area testbeds. It is rare for publications in the systems area to describe which aspects of a problem definition or design were motivated by deployment experience in the wide area. That paucity of examples in the literature shouldn't be confused with testbeds' not leading researchers to such fundamental insights, however. The thesis I will present in this talk is that experience deploying distributed systems on wide-area testbeds routinely leads to fundamental insights of research significance, as measured in new problem definitions and algorithmic solutions to those problems.

3.6 A Formal Model for Network Communication Mechanisms

Martin Karsten (University of Waterloo, CA)

Joint work of Karsten, Martin; Keshav, S.; Prasad, Sanjiva; Beg, Mirza
Main reference M. Karsten, S. Keshav, S. Prasad, M. Beg, "An axiomatic basis for communication," in Proc. of the 2007 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'07), pp. 217–228, ACM, 2007.
URL http://dx.doi.org/10.1145/1282380.1282405

In 2007, a group of collaborators and I presented a paper at the Sigcomm conference that introduced an axiomatic description of network communication. While we have continually received very positive feedback about this work, the ACM digital library shows only 5 proper citations almost 6 years later and in general, there seems to be very little directly related work. Other formal models are typically concerned with more specific problem areas and protocols, such as BGP-style routing. At the same time, it has been surprisingly difficult (for me) to evolve the original communication model into a complete yet simple language that offers tangible benefits. In this talk, I will present recent progress in designing such a formal model/language for basic network communication mechanisms. More importantly, I want to use the opportunity to have a discussion about the usefulness and feasibility of this approach.

3.7 Opportunities and Challenges for Software Defined Network Systems

Wolfgang Kellerer (TU München, DE)

Whereas the Internet has emerged to an economic factor for industries across all disciplines of our information society, its current system architecture fails to support such emerging application requirements in a flexible and dynamic way. In this respect, Software Defined Networking (SDN) marks a fundamental paradigm shift in information and communication networking technology. It introduces an open interface between network hardware realizing data forwarding and the corresponding control software overcoming several limitations of current network architectures. The SDN concept allows for the first time to implement a completely dynamic control of communication networks. Forwarding rules are pushed in runtime from a logically-centralized external control entity to the distributed network hardware. From a network programming point of view, this mechanism allows to adapt the communication infrastructure flexibly and rapidly with respect to changing service demand created by the users of the network. Moreover, the concept of SDN is not limited to the basic switches and routers, but can be viewed as a general concept to increase flexibility and dynamic adaptation in communication networks spanning all network infrastruture including servers and storage.

This presentation reviews the opportunities provided by SDN addressing not only the SDN controller southbound interface (e.g. OpenFlow), but, in particular, the northbound interface towards support for network system applications of heterogeneous application domains. Standardization has already started and many network enterprises expect SDN to solve all problems they have with the Internet. In fact, however, in many respects SDN research is still at its initial research phase with many basic challenges to be resolved.

3.8 Automatic Energy Efficiency Management of Data Center Servers Operated in Hot and Cold Standby with DVDS

Paul Kühn (Univ. Stuttgart, DE)

The operation of large Data Centers with thousands of servers is very costly in terms of energy consumption and cooling requirements. Distributed Data Centers will play a major role in the future Internet for Content Distribution and as Service Delivery Platforms, also known as Cloud Networks. Energy efficiency and cloud network performance are largely reciprocal to each other and have to be optimally managed to save energy costs at one hand and to meet Service Level Agreements (SLA) in terms of throughput and delay at the other hand. Moreover, volatile service requests need a careful load balancing by process/data migration between the distributed Data Centers causing additional costs for data replication and communications. The participating components span a wide field for the management of Distributed Data Centers.

In this contribution a generalized model for resource management of Data Center servers is presented which allows an automatic server consolidation by a load-dependent control of server activations/deactivations using multi-parallel hystereses thresholds, cold and hot server standby, and Dynamic Voltage and Frequency Scaling (DVFS). For the analysis of energy efficiency and performance a multi-server queuing model is defined which is controlled by a Finite State Machine (FSM) and a two-dimensional system state variable which allows for an automatic adaptation to the current system load. The FSM is constructed such that all requirements of the above stated functionalities can be considered and influenced by proper parameterization.

The model can be analysed exactly under Markovian traffic assumptions from which all relevant performance and energy efficiency metrics can be derived. The focus of the presentation is drawn mainly to numerical results for various parametric studies which demonstrate the usefulness of the modeling approach with respect to Data Center Management and shows the tradeoff between the conflicting aims of Energy Efficiency and Performance. The mathematical methodology will be reported in more detail in a forthcoming publication.

An outlook is given to current work on the management of Distributed Data Centers within a Cloud Network operating on the current load or on system-state based process migration between the Data Centers which is controlled by a mapping of virtualized servers on the physical server resources for load balancing maintaing SLA agreements.

3.9 Experimentally driven Networking Research

Jörg Liebeherr (University of Toronto, CA)

Experiments in experimental networking research often follow an engineering design process rather than the classical scientific experiment approach. The different discovery process which involves (in-)validating the realization of a design, as opposed the rejecting/accepting of a hypothesis puts different requirements on experiments. The talk reviews challenges and risks of experiments in the engineering design discovery process.

3.10 SDN++: Beyond Programmable Plumbing

Laurent Mathy (University of Liège, BE)

SDN, as exemplified by OpenFlow, has focused on control plane programmability. While this is a tremendous step forward towards greater networking application flexibility as well as network virtualization, much can also be gained from data plane programmability. This talk explores some whys and hows in this context.

3.11 Congestion Exposure (ConEx) – An Experimental Protocol for the Future Internet

Michael Menth (Univ. Tübingen, DE)

The congestion and delay experienced in the Internet is often caused by a small fraction of heavy users that overuse limited shared resources. Internet Service Providers (ISPs) counteract with measures that are costly or not very effective, impose unnecessary restrictions, or violate network neutrality. This calls for novel congestion management solutions that need input about the congestion state of the network. The IETF is currently developing the Congestion Exposure (ConEx) protocol that reveals the congestion caused by a flow to all nodes on its path. Intermediate boxes may leverage that information for novel congestion management approaches. The talk explains how congestion exposure works, gives example for its use, and illustrates ConEx-based congestion policing in more detail.

3.12 A Virtual Environment for Distributed Systems Research

Paul Müller (TU Kaiserslautern, DE)

Joint work of Mueller, Paul; Cappos, Justin; Schwerdel, Dennis

Networks and distributed systems are an important field of research. To enable experimental research in this field we propose to build a virtual research environment for distributed systems. It should support researchers from various branches of science who investigate distributed systems by providing a virtual environment for their research. With the topology management tool (ToMaTo), a flexible and efficient networking testbed has been built that can be used as a core for the proposed research enviroment. Using various virtualization technologies, ToMaTo is able to provide realistic components that can run real-world software as well as lightweight components that can be used to analyze algorithms at large scale. This paper describes how an additional virtualization technology from the Seattle testbed has been added to ToMaTo to allow even larger experiments with distributed algorithms.

3.13 Software-defined Networks for Distributed Interactive Multimedia Environments

Klara Nahrstedt (University of Illinois – Urbana, US)

Joint work of Nahrstedt, Klara; Ahsan Arefin; Raoul Rivas
Main reference A. Arefin, R. Rivas, R. Tabassum, K. Nahrstedt, OpenSession, "Cross-layer Multi-stream Management Protocol for Interactive 3D Teleimmersion," Technical Report, University of Illinois at Urbana-Champaign, 2013.
URL https://www.ideals.illinois.edu/handle/2142/43348

New Distributed Interactive Multimedia Environments (DIMEs) such as 3D teleimmersive environments are emerging, consisting of multiple 3D video streams, audio, graphics and other sensory information to connect geographically distributed users into a joint virtual space for virtual interaction and remote collaboration. These DIMEs have a very large demand on bandwidth/CPU resources and tight delay requirements for enabling interactivity. Multi-stream, multi-modality and multi-view requirements add to the complexity of the DIME's application model. In the past couple of years, these types of environments have been using extensively the underlying Best Effort Internet infrastructure with bandwidth and delay-aware overlay session-based protocols since underlying IP-based protocols and frameworks such as RSVP/IntServ, DiffServ, or MPLS technologies have not been easily accessible for distributed interactive multimedia application developers (e.g., multiplayer gaming, video-conferencing developers). With Software-Defined Networks (SDN) and their Open Flow implementation, an opportunity emerges for DIME developers to have a much stronger access to lower level protocol stack and QoS control over Internet networks, and hence coordinate the bandwidth and delay controls between session-based and IP-based protocols and layers. The question is 'How' one achieves this cross-layer SDN-DIME coordination and control. In this talk, we will discuss challenges, problems and directions that we see as we explore cross-layer SDN-DIME coordination and control issues in our TEEVE (Teleimmersion for Everybody) testbed.

3.14 Towards Wide-Area Network Virtualization

Panagiotis Papadimitriou (Leibniz Universität Hannover, DE)

Joint work of Papadimitriou, Panagiotis; Dietrich, David; Rizk, Amr; Bozakov, Zdravko; Mathy, Laurent; Werle, Christoph; Bless, Roland
Main reference D. Dietrich, A. Rizk, P. Papadimitriou, "Multi-Domain Virtual Network Embedding with Limited Information Disclosure," IFIP Networking 2013.

The ever-increasing need to diversify the Internet has recently revived the interest in network virtualization. Most virtual network (VN) provisioning techniques assume a single administrative domain and, as such, the deployed VNs are limited to the geographic footprint of the substrate provider. To enable VN provisioning across multiple substrate providers, network virtualization architectures need to address the intricacies of inter-domain aspects, i.e., how to provision VNs with limited control and knowledge of any aspect of the physical infrastructure.

In this talk, we discuss the challenges of multi-provider VN provisioning and present a framework that circumvents the difficulty of VN embedding with limited information disclosure (LID) from substrate providers. We further present a technique for inter-domain virtual link setup based on NSIS.

Given the increasing interest in network programmability and control, we discuss the benefits and challenges of SDN virtualization. In this context, we outline the design of a distributed SDN hypervisor that facilitates the embedding, deployment, configuration and operation of virtual SDNs.

3.15 The NEBULA Future Internet Architecture

Jonathan M. Smith (University of Pennsylvania, US)

URL http://nebula.cis.upenn.edu

The NEBULA Future Internet Architecture is a clean-slate design focused on secure and resilient networking to support cloud computing. It is designed to support applications that would be difficult or impossible today, such as cloud-based real-time control of medical devices. It is comprised of three parts: (1) NCORE, a core network interconnecting data centers; (2) NDP, a new data plane; and (3) NVENT, a new control plane. Two flavors of NDP have been implemented, Icing and TorIP. Icing uses cryptographic markers on each packet for policy enforcement. The proof of consent (PoC) is generated iff every network element agrees with the policy. This has been implemented and integrated with the RapidNet declarative networking system. TorIP ("Tor instead of IP") likewise requires receiver interest, but uses onion routing to defend against malicious ISPs. NEBULA is moving forward rapidly and is looking to deploy over the next two years.

3.16 Multi-Mechanism Adaptation for the Future Internet – MAKI

Ralf Steinmetz (TU Darmstadt, DE)

URL http://www.maki.tu-darmstadt.de

The Collaborative Research Centre (CRC) MAKI (Multi-Mechanism-Adaption for the Future Internet) addresses this challenge. In particular, it investigates all kinds of mechanisms in communication systems, the adaptation, interaction, constant optimization, and evolution thereof. The term mechanism describes both, communication protocols and parts thereof – defining the functionality of communication systems – and the functional aspects of the distributed systems realized on top. We witness a constant development of novel mechanisms. Yet, mechanisms providing equivalent functionality under different conditions coexist, since an adaptation of legacy mechanisms to traffic conditions, bandwidth, etc. is limited. Particularly mobile usage induces highly fluctuating conditions, which would require the online adaptation of the communication system by means of transitions between functionally equivalent mechanisms – which is mostly impossible as of today. Interactions between mechanisms that jointly depend on each other are even more complex and require coordinated transitions in groups of equivalent mechanisms, so-called multi-mechanism adaptation.

3.17 Some Reflections on Experimentally driven Research and Testbeds in Future Network Research

Phuoc Tran-Gia (Univ. Würzburg, DE)

During the past decade we witness numerous activities towards next generation network. A huge portion of them deals with building generic testbeds and experimental platforms, with a number of experimental facilities initiated in several countries, e.g in the US, EU, Japan etc. The talk gives some reflections on the pros and cons of such ventures and concludes with a view on how the current development of test methodologies emerges.

3.18 Overlay Networks as Innovation Engines

Oliver P. Waldhorst (KIT – Karlsruhe Institute of Technology, DE)

Joint work of Waldhorst, Oliver P.; Zitterbart, Martina

The current Internet architecture has proven to be quite stable due to the invariants imposed by IP. Innovations are mainly seen in new networking technologies on the lower layers and in applications, i.e., on the very bottom and top of the architecture, respectively. Nevertheless, novel applications on the one hand impose new requirements on the Internet, e.g., robustness, security, and mobility support, and call for sophisticated network services, e.g., group communication or in network data processing. On the other hand, applications have to be ready to be deployed on top of evolving networking technologies leading to potentially very heterogeneous networks. In this talk, we argue that application layer overlay networks can support an application developer in coping with these challenges. We introduce the Spontaneous Virtual Networks (SpoVNet) architecture as an illustrating example. Beyond what has been achieved with overlays, we outline steps towards our vision of generating and deploying a networked application automatically from a high level specification.

4 Working Groups on Prescriptive Network Theory

4.1 Group A

Ruben Cuevas-Rumin (Universidad Carlos III de Madrid, ES)

Joint work of Bozakov, Zdravko; Crowcroft, Jon; Cuevas-Rumin, Ruben; de Meer, Hermann; Müller, Paul; Stingl, Dominik

The discussion focused on two main aspects:

First, we discussed what is the most appropriate approach to design a future Internet architecture. The research effort to define the future internet architecture should combine short and long term approaches. In particular, the short term approach is an evolutionary one that improves or extends the current architecture to address new requirements in the short term. It must take into account current technical as well as external (e.g., business model or polices) constrains. Instead, the long term approach should present a clean state design that should not consider current external constrains (e.g. the current business model of Content Providers may very well change in few years from now). Furthermore, the definition of a new architecture should come from a "new point of view". A possible "point of view" may be the Software Engineering discipline that provides more flexibility.

Second, we discussed few important requirements that need to be consider in the design of the future Internet Architecture:

- Privacy is a new requirement that has gained relevance in the last years associated to the 'boom' of OSNs. A future Internet architecture should be able to provide privacy as a service.
- A power-aware architecture is needed. At the current growth of bandwidth, processing capacities, etc. the current infrastructure will not be able to dissipate the generated heat. Then, we need to take in consideration power consumption as a design requirement of the future Internet architecture.
- High availability: the current Internet guarantees a decent level of availability for a non critical service, however the availability guarantees offered for critical services is deficient. The Future Internet architecture has to deal with this issue and offer a very high availability.

As summary of the discussion, we concluded that any new architecture should provide evolvability/flexibility/adaptiveness and therefore self-organization/autonomic operation is a key necessity.

4.2 Group B

Markus Fidler (Leibniz Universität Hannover, DE)

Joint work of Fidler, Markus; Gross, Christian; Hofmann, Markus; Pries, Rastin; Sarrar, Nadi; Tran-Gia, Phuoc; Mühlhäuser, Max

Group B discussed what we would refer to as the 'meta level' of prescriptive network theories, where we focused on:

1. the different means for construction of (Future Internet) networks, not only theories;
2. the spectrum of interesting issues (foci) that such means could concentrate on;
3. the possible degree of generality and of breadth.

With the goal of a formal theory in mind, we questioned whether there may be any fundamental laws in networking that lay out the foundation of such a theory, like the laws of nature in physics. During the discussion a number of examples of technical systems came up where basic constraints, rules, and/or theories exist that have greatly advanced the respective fields and in some cases have even been used in a 'generative way.' Examples include Shannon's capacity, Erlang's formulas, e.g., for dimensioning of telephony switches, and the AIMD-law in TCP congestion control.

Regarding networking, we first clustered the problems/fields that we want to address and based on this we formulated the kinds of means that we want to have. The fields of interest are (without claim to be complete):

1. high level
   - scalability, evolvability
   - divide & conquer vs. (inevitable) interdependencies
2. functional
   - multicast
   - multihoming
   - mobility
3. non-functional
   - energy efficiency
   - security, privacy, quality of service (bandwidth, latency, fairness, etc.)
   - economy

In the following discussion, it came out that there are numerous formal theories and models that can provide a deep understanding how and in which way certain mechanisms work, e.g., queueing theory, graph theory, game theory, and many others. These methods are frequently used to analyze non-functional aspects. The question how to engineer a certain function resorts much less to theory and rather relies on cookbooks of known mechanisms that exist, e.g., for medium access control, ARQ, congestion control, and so on. While many non-functional aspects of these methods are well-understood, e.g., the throughput of ALOHA vs. CSMA, we seem to lack a generative theory that enables finding new mechanisms. Finally, considering network architecture on a high level, we face problems that are hard to formalize and hard to anticipate so that we are confined to guidelines and best practices. In conclusion, we formulated how formal we expect a prescriptive network theory to become with respect to the complexity of the problem:

Computing a solution: optimizing a protocol, naming scheme, etc.;
Engineering a solution: designing a custom network, e.g., WSN, enterprise, etc.;
Crafting a solution: designing *the* Future Internet.

4.3 Group C

Florin Ciucu (TU Berlin, DE)

Joint work of Ciucu, Florin; Hausheer, David; Hollick, Matthias; Karp, Brad; Karsten, Martin; Mathy, Laurent; Santini, Silvia

The group questioned existential aspects of a generic 'Formal Theory' such as the necessity of explaining vs. designing the 'reality', or its temporal positioning. One can in particular question the usefulness of a theory either retrospectively (as being meant to 'explain' what has been done so far) or prospectively (as being meant to help what has not been done yet). The group also addressed questions related to its limits and scope, and also attempted to define benchmarks for its usefulness. For instance, in the context of a hypothetical network theory, one should consider temporal aspects such as design time or the frequency of break downs.

One interesting analogy that emerged during the discussion is that while physicists take for granted inherent laws that nature obeys, network scientists are confronted with a lack of uniform laws since artificial entities are by default subject to many unknown and uncontrollable inputs. Moreover, while nature has inevitably led to unfortunate 'designs', those have been corrected by evolution; in turn, the role of 'evolution' in the context of artificial entities remains elusive.

The group then debated about the appropriateness of defining a 'prescriptive theory' in either a strict sense (i.e., axiomatically driven) or more generally by considering in particular conducive facets for system design. The follow-up discussion assumed the existence of such a theory, whose axioms could be conceivably applied to building a network, but which eventually would turn out as bogus. This apparent contradiction was resolved by arguing that an 'accurate' theory can only produce a single network, in which case 'restrictive' would be semantically more appropriate than 'prescriptive', as a qualifier for 'theory'. The rather contrived line of thought that emerged was followed by an enlightening contrast between architects who can both follow and break design principles, and network scientists who are confronted at the very least with the lack of specifications for network input.

In attempting to exemplify success stories of prescriptive theories, the group first mentioned control theory and its key role to transport protocols, meta-routing, or XCP. The initial consensus was that control theory is only partly prescriptive and that it is restricted by the need of making many simplifications and assuming complete knowledge. In an effort to underline the challenges faced by network scientists, it was humorously speculated that control theorists believe that 'everything' (including networks) can be controllable or at the very least explained. The second example was game theory which was thought to be as being more prescriptive than other theories by arguing that it starts out with the axiom that players act in their own interest. However, by reflecting on applications to selfish routing, the prescriptive facet of game theory started being questioned by considering the position of ISPs. Queuing theory was the last example, and that was argued to be characterized by descriptive rather than prescriptive aspects since it is based on axioms of mathematics.

Another separate line of discussion concerned broken aspects and possible fixes in the current Internet. Immediate facts being mentioned included the existence of too many protocols for the same purpose, the network overloading with functionality, the lack of bounded latency, the apparent impossibility of building proper firewalls, the lack of extreme availability and reliability, or the unclear role of admission control. A possible cause for the current frustrating state of the Internet was identified to be that functionality itself is indistinguishable from the infrastructure. More concretely, the whole network appears to look like an embedded system, conceivably implying that who knows the infrastructure also knows the functionality. In plastic terms, the Internet was compared to a washing machine, and it was further recommended to adopt similar principles as the airline industry, i.e., immediate redesign after failure.

The discussion overall spurred some meta-thoughts which apart from being plausible certainly deserve being reflected upon:

- All theories are limited in scope and also in time.
- Internet research is mostly empirical.
- Technology changes much faster than theorists can adapt.
- We need evolution theory to model the Internet.
- Lets just accept there will not be a clean-slate Internet.

4.4 Group D

Tobias Hoßfeld (Universität Würzburg, DE)

Joint work of Dietrich, David; Hoßfeld, Tobias; Klug, Karl; Nahrstedt, Klara; Papadimitriou, Panagiotis

This report documents the discussions and outcomes of the working group D with the participants above during the Dagstuhl seminar on "Future Internet". Thereby, the group agreed to discuss on a methodology towards prescriptive network design and had a good mixture between researchers from industry and academia.

The term "prescriptive network design" reflects a theoretical approach to design a full-service, comprehensive network close from scratch based on a set of constraints. As a result of the group discussions, five basic steps are to be fulfilled for prescriptive network design (PND). In particular, the constraints for network design have to be de-constrained in order to derive a minimum required functionality and an appropriate function split. As a pre-requisite, the elements of PND have to be identified, while the actual constraints and requirements are derived for a set of networks, use-cases, and models which are stored in a common database. Then, an interface between prescriptive and descriptive network design has to be defined in order to allow a flexible and adaptive solution (that may be seen as an iterative process where 'top-down' and 'bottom-up' are converging to a pragmatic, near-optimal solution). Thus, the recommended methodology for PND is as follows.

- Step 1. What are the elements of prescriptive network design?
- Step 2. Database of networks, use-cases, models, etc.
- Step 3. Minimum function split
- Step 4. Interface between pre-/descriptive
- Step 5. Recommendations from theory

Step 1 addresses the elements of PND. It has to be derived which are the basic elements, what are the properties of the elements, the capabilities of the architectural elements, and which are the required fundamental services. Thereby, it can be distinguished between a) functional and b) non-functional elements, constraints, or requirements. An example of a functional element is the negotiation of resources, quality guarantees, security, etc. Thereby, the location of functions has to be considered in PND, too. Non-functional elements address for example scalability. However, it has to be clarified which are the basic elements of PND, e.g. protocols, topology, switches, etc.

Step 2 recommends a database of networks, use-cases, models, etc. In particular, the requirements for those use-cases are to be derived and collected. This includes non-functional and functional constraints or requirements, as well as business model constraints or requirements. In general, models for different use-cases, models for network dynamics, etc. have to be taken into account in this database. A concrete use-case is for example an educational campus with changing traffic demands according to the students habits and living on the campus.

Step 3 aims at de-constraining the constraints and tries to derive the minimum function split. This minimum function split reflects the smallest denominator in order to compose higher functions and services, i.e., it is the basis for functional composition. In particular, a classification of the constraints and requirements from the use-cases is to be developed. Then, the function split of the PND is based on this classification. Thereby, the minimum number of functions is to be derived, but the complexity of functions has to be considered too.

Step 4 of the recommend methodology is the definition of an interface between prescriptive and descriptive network design. Since the different services know best their requirements, it seems to be straightforward that the services describe their requirements themselves. The prescription is implemented via an interface between different services in order to design the network from a holistic point of view for all services. This interface allows an adaptive and flexible network design.

Finally, Step 5 of the PND are recommendations by the theory. In particular, the theory of PND should answer questions like the following. “How to design the network protocol stack?” “What is an appropriate network topology for the basic elements?” “How shall concrete mechanisms, e.g. schedulers, look like?” As an outcome of the PND theory, tools for network planning but also for network operation may be derived. This is in particular relevant in the context of network virtualization and the different stakeholders like physical infrastructure providers, virtual network provider, and virtual network operator. From a business point of view, use-cases may be defined with prescriptive description of networks, which lead to valid business cases for network operators.

4.5 Group E

Oliver P. Waldhorst (KIT – Karlsruhe Institute of Technology, DE)

Joint work of Godfrey, Philip Brighten; Kühn, Paul; Liebeherr, Jörg; Menth, Michael; Smith, Jonathan; Waldhorst, Oliver P.

Motivated by the statements of the seminar participants in the opening session the group selected ‘latency’ as an example requirement for future Internet architectures and discussed reasons for latency as well as ways of reducing it using architectural and methodical approaches. The group identified transmission / propagation delay, switching delay, queuing delays, and security handshakes as main sources of latency. While some of these sources cannot be changed, e.g., propagation delay is bounded by the speed of light or handshakes cannot be avoided without sacrificing security, some other can be tackled by changes architecture or paradigms. This holds in particular for queuing delays introduced by router buffers.

Architectural changes for reducing queuing delays include a circuit switching architecture without any buffers, which means trading delay for loss, or losing the statistical multiplexing gain. Obviously, controlling latency is expensive, since it requires sophisticated control mechanisms. Another way to reduce latency in general is multiplexing resource usage across multiple resources, for example by requesting a web site from multiple servers and using the first response. From a methodical point of view, a formalism is required to compare one architecture for reducing latency to another. One example is the enhancement of the Axiomatic Basis for Communication proposed by Martin Karsten et al. with means to compute latency.

5 Podium Discussions

5.1 Experimentally driven research

Tobias Hoßfeld (Universität Würzburg, DE)

On the second day of seminar, experimentally driven research was one of the major discussion topics. To this end, five different talks were given to stimulate the discussions among the participants:

- Martina Zitterbart and Oliver Waldhorst: "overlay networks as innovation engines"
- Jörg Liebeherr: "experimentally driven networking research"
- Brad Karp: "wide-area distributed system deployments yield fundamental insights"
- Markus Hofmann: "<provoke>research testbeds considered harmful</provoke>"
- Phuoc Tran-Gia: "some reflections on experimentally-driven research and testbeds in future network research".

Finally, a podium discussion with the speakers above took place in order to raise questions to the speakers and to express own opinions about experimentally driven research.

Just before the podium discussion, Phuoc Tran-Gia gave a talk and shared his reflections on experimentally-driven research and testbeds in future network research. Thereby, the following five key observations and statements were made by Phuoc Tran-Gia.

1. Analytical studies are necessary for future Internet research. A recommended structure of experimental-driven research projects includes two different facets that are (A) research (sub-)projects and (B) the experimental facility. On one hand, the research (sub-)projects (A) are needed to investigate particular questions in the context of future network research. This does not necessarily mean that those research projects need to be tested within an experimental facility, but the answer to those research questions may also be derived from theoretical studies, i.e. analytical approaches or simulations. One particular future internet research question addresses scalability which requires in particular simulations and analytical models to overcome limitations of testbeds in size. On the other hand, the experimental facility (B) is to be developed. Sometimes this facility is then used by the research projects from (A). This structure of experimental-driven research was successfully applied in the G-Lab project combining future internet research and experimentation, whereby the experimentation consumed 10-15% of the overall efforts in the G-Lab project. As a prerequisite for experimental-driven research projects, test cases and test interfaces in the experimental facility need to be defined. Only parts of the developed solutions in the research projects (A), e.g. some virtualization mechanisms, may then be tested

later within the experimental facility (B). Hence, successful experimental-driven research needs a combination of measurements, simulations, analysis.

2. Generic experimental platforms have the following advantages. They enable large-scale experimental platforms which can be reused for several test-cases. This reduces the setup time for the testbed initiation phase. Further, the generic experimental platform provides a clean environment which allows to generate reproducible results. The reproducibility of the experiments is however questionable. Another advantage of generic experimental platforms is adaptability which allows to enlarge a number of nodes in the test phase easier, to quickly change test images in the platform, to enable federation with testbeds, etc. Finally, generic experimental platforms can in principle be designed for sustainability.
3. Major problems with generic experimental platforms however outweigh the advantages. Often, generic experimental platforms cannot be used in most emerging test cases, as they need e.g. special requirements not offered by the platform. The lifetime of generic experimental platforms is limited due to the hardware and software lifecycles. Further, the question is raised how generic is a generic testbed, how to do scalability analysis, whether experiments are really reproducible. The question arises how "real" are experimental platforms? Conducting measurements e.g. in Planetlab and via crowdsourcing show that the obtained results are not representative. Another major drawback of generic experimental platforms is the fact that the focus is often only on the testbeds itself while classic methods are often neglected. For example, testbeds can only partly support overload-preventive design process and hardly support scalability analysis. Another example is network-aware application design which makes testbeds more complex. Thus, it is an open question whether application-aware network design & network-aware application design benefit from testbeds.
4. Some observations on generic testbeds were stated. First, most testbeds are underutilized, many testbeds are not used by industries and there is a lack of users and experimenters of the testbed. Another observation is that generic testbeds start to break down in several disjoint testbeds. Further, maintenance costs for the testbed are significant and large efforts have to be spent for sustainability and life-time of the testbed.
5. Possible solutions to overcome the limitations of generic experimental platforms are the following. First it is important to design research projects together with experimental facility research (see statement 1. above). This includes to design and finance test projects together with the testbed setup. Further, software-defined experimental platforms are easy to extend and to change and may be combined with other techniques. For example, the human cloud and real users may be integrated in experimental facilities, e.g. by means of crowdsourcing. The integration of the human cloud is sometimes required, e.g. when looking at Quality of Experience (QoE), and cannot be offered by a machine cloud. Another recommendation is the federation and integration of special testbeds in generic experimental platforms in order to extend their capabilities and possibilities.

Directly after the presentation by Phuoc Tran-Gia, the podium discussion took place with the speakers on experimental-driven research in front of the audience: Martina Zitterbart, Oliver Waldhorst, Jörg Liebeherr, Brad Karp, Markus Hofmann, Phuoc Tran-Gia (in chronological order of presentations).

The first question to the panel was raised by Tobias Hoßfeld: "Is real user behavior important in testbeds? Do we have to include real users in testbeds? Can crowdsourcing help to identify problems like signaling storm?" There was, however, no agreement in the panel and in the audience about the integration of real users in testbeds. Jörg Liebeherr mentioned that the purpose of a testbed is to find limits of technologies, e.g., security features of cars are not tested using real users but dummies. In contrast, Phuoc Tran-Gia mentioned that the car example does not hold as other features are tested in fact by real users. Further, Markus Hofmann explained that real-world problems are overseen without real users. To this end, Brad Karp mentioned that there is a continuum between basic research and experiments and that crowd testing and machine testing are important. Another viewpoint was given by Oliver Waldhorst, as the application developer (as a real user) has to be considered in experiment-driven research, too.

The second question was asked by Matthias Hollick and focused on the presentation by Markus Hofmann: "Do you provide a testbed for application developers?" Markus Hofmann explained that this is not done by their company, as they are only considering the requirements, expectations, and so on. Nevertheless, there is a change in research, as customers have to be asked about scenarios. Further, it is hard to predict the consequences of application development. In contrast, e.g., to car industry, scalability is an important issue in future networking research.

Then, the discussion focused on scalability. Phuoc Tran-Gia mentioned that scalability analysis requires test theory, analytical models, etc.; it cannot be tested in testbeds. Scalability is a big challenge, because 1 Mio testing devices is not possible in a testbed, although it can be extended to a certain extent by crowdsourcing. Jon Crowcroft asked what will change when you test a system with 10 Mio users. Will unknown effects happen in large-scale, e.g., synchronization effects? Klara Nahrstedt mentioned the example of group dynamics from social science. In particular, group dynamics change depending on size of group and a change is observed between 1, 5, 10, 50, 100, 150 users, while above 150 users no additional changes happen. Jörg Liebeherr criticized that any scalability study leaves the reader unsatisfied, e.g., due to model assumptions. However, Paul Müller argued as mathematician that we can believe in 'small numbers' and draw conclusions for scalability analysis, since there are statistical methods to investigate scalability and to provide valid results. Markus Hofmann finally mitigates the discussion on scalability analysis, as it is not a binary decision to have large-scale experiments or not, since we need all different methods in research. It will not help to use just one or the other.

Paul Kühn stimulated the discussions then in two different directions. First, the effects of social networks have to be considered in future Internet research which was commented by Jon Crowcroft that social networks and communication networks can be combined. Second, errors in software deployment and error propagation in software-defined networking need to be considered in (SDN) experiments. When considering Software Defined Networking, we have to deal with a lot of software errors. This will be problematic, because we first have to identify software problems. Brad Karp answered that when looking at Cisco, we have millions of lines of code, which we cannot take a look at. This will not be the case with SDN. However, Paul Kühn argued that the frequency of changes in software is a lot higher in SDNs. Paul Müller mentioned that a formal verification of software and tools for software verification are required, which is a general software problem, but tools are partly available as pointed by Brad Karp.

Next, Phuoc Tran-Gia asked the following: "Is reverse-engineering of (human-designed and implemented) mechanisms science or engineering? Example: YouTube re-engineering of mechanism ... and then the mechanism is changed with the next version." This was commented by an additional question by Paul Müller: "Is academia running behind industry, e.g., Skype?" However, Klara Nahrstedt noted that there are a lot of success stories coming out from university and academia. Markus Hofmann mentioned that industry does not care about optimal solutions, but pragmatic solutions. Industry often gets along with a non-optimal version, while university often tries to find the optimum.

The next question was raised by Klara Nahrstedt: “Heterogeneous devices are not supported by testbeds. How to plugin more realistic traffic patterns? How to bring applications closer to network testbeds?” Paul Müller mentioned the Seattle tool for this, while Phuoc Tran-Gia also stated that many testbeds are ‘too low’ and neglect applications and users.

The last question in the podium discussion was queried by Johnathan M. Smith: “How to sample and privatize data?” There was a lively discussion on sampling and privacy. Markus Hofmann answered that a large amount of data is available for research, but often not accessible due to privacy concerns, regulations, and so on. He additionally asked: “Can we make providers (like YouTube) share data?” and claimed that sampling and anonymization tools are required that are broadly acceptable. Even within companies, it is difficult to get data from other groups. Brad Karp agreed that anonymization is a hard problem and that it is unclear how to ensure anonymization of data.

5.2 Use cases of SDN

Jon Crowcroft ((University of Cambridge, UK))

On the last day of the seminar, we had a session where we tried to enumerate some of the use-cases for SDN, and this is a partial list of what was captured:

1. multi tenant data centers have a need for specialised networking for different tenants who may have different network service requirements – just picking a few examples, one tenant might need multicast for serving IPTV, while another might need delay bounds for traders or gamers, whilst still another might need in-network processing to support better MapReduce task deterministic throughput.
2. Multi User VR and Massive Multiplayer Online Role Playing Games have large scale dynamic network requirements whose parameters keep changing – one could certainly imagine supporting this partly through SDN.
3. Live Migration of Virtual Machines is increasingly used both for load balancing and for high-availability/persistence during scheduled maintenance. VM Migration puts a spike load onto the net, and could interfere with routine steady state traffic unless serviced specially, and requires itself, specialised support for continued access (re-routing existing flows, and redirecting new clients, for example). This could be supported as an SDN function.
4. The Internet of Things (IoT) is touted as a big driver for networking (e.g. IPv6 deployment to support the massive increase in globally reachable devices). IoT will also require specialized network functionality (securing access, delay bounding access for some sensor/actuator cyber-physical applications, and many as yet unforeseen applications that will no doubt emerge).
5. Content Distribution Networks (CDN) need management. Some CDNs place difficult load on an ISP – especially CDNs that are all or partly P2P, perhaps interfering with peering arrangements. Managing this could easily be seen as a good match to SDN.
6. Middlebox management is sorely needed. Middleboxes increasingly are the main cause of ossification of the network – the inability to deploy new transport layer protocols or extensions is mainly traceable to the ad-hoc mature of the large number of different

middle box functions deployed throughout the Internet. Bringing these into a coherent framework would allow some semblance of progress to be made. SDN again would be the tool to replace the ad-hoc functionality with a clean slate programmable system with a public, open standard API.

7. This is the metaclass of the SDN use case – management of multiple SDNs will itself be a challenge.
8. Hybrid clouds where the cloud supports a set of user applications and the set of SDN apps is an interesting case of meta-management. Co-existence of the applications and the customised, specialised support for these applications in the software defined network is a key requirement, for example, in today's' data centers.
9. Resilience in networks currently relies on ad-hoc approaches to providing replication. SDN can help unify a set of mechanisms under one control plane.
10. Cross-layer design of distributed applications requires potentially more open, possibly reflective APIs in SDN, so that the Application and the SDN can co-evolve efficiently.
11. Enterprise infrastructure setup is a key need in large scale private intranets. There are many such systems in the world, and often their owners incur high costs to provide customised network services. SDN offers a way to build more flexible networks that could be matched to an enterprise's needs more as a matter of configuration than bespoke engineering.
12. Improved security may be on the cards if SDN takes on board improved software practises, using safer programming languages, and trusted computing bases, and techniques for information flow analysis, software verification, and so forth.
13. One simple hope for SDN might be to take research results in policy routing (e.g. meta-routing) and do a one-time replacement of BGP.

We also considered SDN in data plane.

1. Data plane middleboxes already interfere with TCP/IP packets in an unstructured way (frequently, to improve operation of protocols in wireless networks such as 3G and 4G nets, but also interfering with the ability to deploy new versions of TCP. SDN could include data plane packet processing, at least near the edges of the network where performance requirements can be met affordably.
2. Fine grain media control could be another SDN data plane activity – e.g. video and audio re-coding for different receivers with different rendering capabilities.
3. Network as a Service (NaaS) for data center (e.g. mapreduce) in-net application code has also been suggested as an SDN data plane task. The TCP incast problem can be solved by processing a fixed number of shuffle phase data packets in switches, with relatively simple tasks.

In addition to numerous use-cases or applications on top of SDN, SDN was also found to have the potential to connect technologies below the SDN implementation. But there are broader questions that we leave unanswered here:

1. Can we use SDN to Connect IP and non-IP networks?
2. Could we do layer 2 and layer 4 SDN via OpenFlow?
3. What could make SDN harmful?
4. What are the key SDN business cases?

6 Seminar Programme

Monday	Prescriptive network theory
09:00-09:30	Welcome and general introduction
09:30-10:30	One minute madness: introduction of participants & Future Internet statement Scribes: Zdravko Bozakov and David Dietrich
11:00-12:00	Markus Fidler: introduction to prescriptive network theory Martin Karsten: a formal model of network communication mechanisms Scribes: Christian Groß and Dominik Stingl
14:00-15:15	Ralf Steinmetz: multi-mechanism adaptation for the future Internet Jon Crowcroft: prescriptive network theories Scribes: Christian Groß and Dominik Stingl
15:15-15:30	Definition of topics for group work and opinions Scribe: Zdravko Bozakov
16:00-17:45	Breakout sessions in group work
19:00-20:00	Wrap-up of group work results in plenum

Tuesday	Experimentally driven research (in overlays)
09:00-10:15	Martina Zitterbart/Oliver Waldhorst: overlay networks as innovation engines Jörg Liebeherr: experimentally driven networking research Scribe: Nadi Sarrer
10:45-12:00	Brad Karp: wide-area distributed system deployments yield fundamental insights Markus Hofmann: <provoke>research testbeds considered harmful</provoke> Scribe: Florin Ciucu
13:45-14:20	Phuoc Tran-Gia: some reflections on experimentally driven research and testbeds in future network research Scribes: Rastin Pries and Tobias Hoßfeld
14:20-15:10	Podium discussion Scribes: Rastin Pries and Tobias Hoßfeld
16:00-17:45	Paul Müller: a virtual environment for distributed systems research Jonathan M. Smith: NEBULA future Internet Michael Menth: Conextion Exposure (ConEx) – an experimental protocol for the future Internet Paul Kühn: automatic energy efficiency management of data center servers operated in hot and cold standby with DVDS Scribes: Panagiotis Papadimitriou and Ruben Cuevas-Rumin
17:45-18:15	Discussion and opinions

Wednesday	SDN, virtualization, and OpenFlow
08:30-10:00	Wolfgang Kellerer: opportunities and challenges for Software Defined Network systems Klara Nahrstedt: Software Defined Networks for distributed interactive multimedia environments Laurent Mathy: SDN++: beyond programmable plumbing Scribe: David Dietrich
10:30-11:00	Panagiotis Papadimitriou: towards wide area network virtualization Scribe: David Hausheer
11:00-12:00	Discussion, opinions, and use cases of SDN Scribe: David Hausheer
12:00-12:15	Seminar resume and farewell

Participants

- Zdravko Bozakov
Leibniz Univ. Hannover, DE
- Florin Ciucu
TU Berlin, DE
- Jon Crowcroft
University of Cambridge, UK
- Ruben Cuevas-Rumin
Universidad Carlos III de Madrid, ES
- Hermann de Meer
Universität Passau, DE
- David Dietrich
Leibniz Univ.t Hannover, DE
- Markus Fidler
Leibniz Univ. Hannover, DE
- Philip Brighten Godfrey
Univ. of Illinois – Urbana, US
- Christian Gross
TU Darmstadt, DE
- David Hausheer
TU Darmstadt, DE
- Markus Hofmann
Alcatel-Lucent Bell Labs – Holmdel, US
- Matthias Hollick
TU Darmstadt, DE
- Tobias Hoßfeld
Universität Würzburg, DE
- Brad Karp
University College London, UK
- Martin Karsten
University of Waterloo, CA
- Wolfgang Kellerer
TU München, DE
- Karl Klug
Siemens – München, DE
- Paul J. Kühn
Universität Stuttgart, DE
- Jörg Liebeherr
University of Toronto, CA
- Laurent Mathy
University of Liège, BE
- Martin Mauve
Heinrich-Heine-Universität Düsseldorf, DE
- Michael Menth
Univ. Tübingen, DE
- Max Mühlhäuser
TU Darmstadt, DE
- Paul Müller
TU Kaiserslautern, DE
- Klara Nahrstedt
Univ. of Illinois – Urbana, US
- Panagiotis Papadimitriou
Leibniz Univ. Hannover, DE
- Rastin Pries
VDI/VDE Innovation + Technik GmbH – München, DE
- Ivica Rimac
Bell-Labs Alcatel-Lucent, DE
- Silvia Santini
TU Darmstadt, DE
- Nadi Sarrar
TU Berlin, DE
- Jonathan M. Smith
University of Pennsylvania, US
- Ralf Steinmetz
TU Darmstadt, DE
- Dominik Stingl
TU Darmstadt, DE
- Phuoc Tran-Gia
Universität Würzburg, DE
- Oliver P. Waldhorst
KIT – Karlsruhe Institute of Technology, DE
- Klaus Wehrle
RWTH Aachen, DE
- Michael Zink
University of Massachusets – Amherst, US
- Martina Zitterbart
KIT – Karlsruhe Institute of Technology, DE