Chapter 1

Introduction

1.1. Evolution of application and network layers

The accelerated development of the Internet and the multitude of networked mobile devices (e.g. smartphones, personal digital assistants (PDAs), tablets, netbooks and laptops) have facilitated the development of a vast family of distributed multimedia applications such as Video on Demand (VoD), video-conferencing, Internet Porotocol Television (IPTV) and Voice over IP (VoIP).

Requirements and preferences of these applications have become very complex when compared to traditional downloading, Web-browsing or e-mailing first generation of distributed applications, for which a reliable and ordered transport service (such as the sevice offered by the traditional Transmission Control Protocol (TCP)) operating over a wired Best-Effort network service was quite well adapted.

However, this new generation of distributed multimedia applications presents more complex requirements of Quality of Service (QoS), mainly expressed in terms of time (e.g. end-to-end delay, multimedia synchronization and jitter), bandwidth (e.g. high and variable bandwidth) and reliability (e.g. tolerance for partial reliability and partial order) requirements.

In past years, several initiatives have been carried out to enhance the basic Best-Effort network service in order to provide new QoS-oriented service models (e.g. DiffServ and IntServ). Moreover, new technologies providing high-speed, wireless and mobile network services have deeply modified the QoS characterization of the network layer thus leading to a more complex service model in terms of bandwidth, losses, delay or jitter.

Furthermore, networked applications running on mobile devices are exposed to an even more complex service model due to the dynamicity of perceived QoS when moving from high-speed and high-bandwidth networks (e.g. ADSL networks at home) to variable bandwidth and high delay networks (e.g. when operating over WiFi or 3G mobile wireless networks).

This important evolution of application and network layers has deeply impacted the traditional transport layer. Indeed, traditional transport protocols (i.e. TCP and User Datagram Protocol (UDP)) were well dimensioned to the original Best-Effort network model. However, specializations of transport mechanisms have been required to cope with new network technologies (e.g. TCP extensions for satellite or WiFi networks).

Likewise, new protocols such as Datagram Congestion Control Protocol (DCCP), Stream Control Transmission Protocol (SCTP) and Multipath Transport Protocol (MPTCP) have been proposed to enhance the service offered by traditional protocols in order to provide new specialized transport functions (e.g. more adapted network congestion avoidance strategies, multihoming support for mobility or multipath support for devices integrating multiple network interfaces).

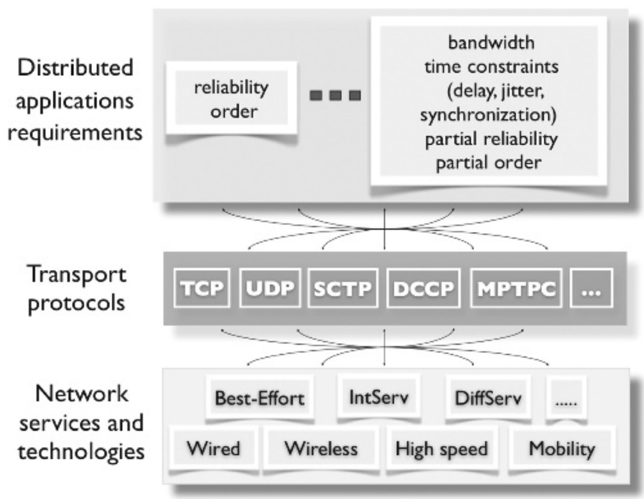

Figure 1.1. *Problem context*

Figure 1.1 summarizes the application, transport and network layer evolution and illustrates the complexity involved in providing the adequate adaptation service at the transport layer.

Even if the transport layer evolution represents a classical example of software change, not much effort has been invested in applying advanced software engineering practices aimed at preparing the basis for the new extensions and specializations that will certainly be required in the future.

Based on this experience, it is realistic to anticipate that an adequate model-driven engineering methodology would facilitate the design of a flexible architecture providing the required extensibility and reusability capabilities in order to incorporate future protocol extensions and specializations, also adapted to the diversity of applications, network services and user devices.

Moreover, new software architecture paradigms enabling service characterization and dynamic service discovery, selection, composition and deployment should be formally integrated into such modern transport layer architecture.

Furthermore, an adequate framework enabling the design of self-adapting transport services in dynamic and heterogeneous network environments, in order to satisfy a large diversity of application requirements, should also be incorporated within such architecture in order to develop an extensible next-generation transport layer.

This book presents a set of iterative and incremental solutions aimed at defining new protocols and applying a specialized software engineering methodology able to integrate this evolution of requirements and network services in order to implement well-adapted transport services. This methodology based on Unified Modeling Language (UML)-based and ontology-based models and integrating service-oriented, component-based and autonomic computing approaches is intended to design the next-generation transport layer.

1.2. Summary of contributions

This book is the result of several years of research in the area of transport protocols and proposes a well-experimented methodology and a set of fundamental approaches and paradigms.

The methodology proposed is based on a model-driven architecture (MDA) design, initiated by abstract standard models of the transport layer and guided by the trends and evolutions of transport protocols specifications and implementations. The resulting model is implemented by an ontology incorporating the semantic related to the services, protocols, functions and mechanisms of the transport layer. Likewise, an incremental and iterative design approach based on the use of the UML language will be used to analyze, design and validate standard and specialized transport mechanisms and functions.

Component-based and service-oriented architecture approaches represent a first paradigm guiding the transport layer design. Based on the semantic model of the transport layer, a service-oriented architecture able to dynamically offer the most adequate transport service based on the application requirement expressions and on the available transport protocols and network services can be designed. Likewise, a component-based transport layer architecture enabling the dynamic composition of reusable transport mechanisms based on the semantic associated with the expected final service can be dynamically composed and used.

The second paradigm represented by the autonomic computing approach is related to the fundamental adaptation function that needs to be guaranteed by the transport layer. While the first paradigm focuses on the adequate selection and composition of transport protocols and mechanisms, the autonomic computing approach is aimed at providing a framework to design and develop the adequate adaptation functionalities to be provided by the transport protocols during the data transfer phase in order to cope with the dynamicity of the service offered by the network layer.

An autonomic computing paradigm can be achieved by considering two maturity levels: adaptation and self-adaptation. The first level consists of providing mechanisms that can be parameterized or reconfigured in order to satisfy new requirements or to cope with changes in the context. The second level consists of integrating an autonomic entity able to analyze the service being provided and to decide the required corrective adaptation actions (i.e. self-adaptation) in order to respond to changes on requirements or the context.

Based on the proposed methodology and integrating the introduced approaches and paradigms, this book proposes a well-proven engineering process to design and develop evolving and smart software architectures. These are the basis of the proposed guidelines aimed at designing and developing an ontology-driven, component-based, service-oriented and autonomic computing architectural framework intended to be integrated within the next-generation transport layer.

Lessons learned from the applied methodology and from research work on component-based and adaptive transport protocols as well as several perspective studies to be carried out in order to design and develop self-managing (i.e. discovery, selection, composition, deployment and adaptation) autonomic properties of the next generation of transport protocols are also presented.

1.3. Book structure

Figure 1.2 represents the book structure and summarizes the iterative and incremental design and development of transport protocols as well as the methodology, frameworks and paradigms proposed.

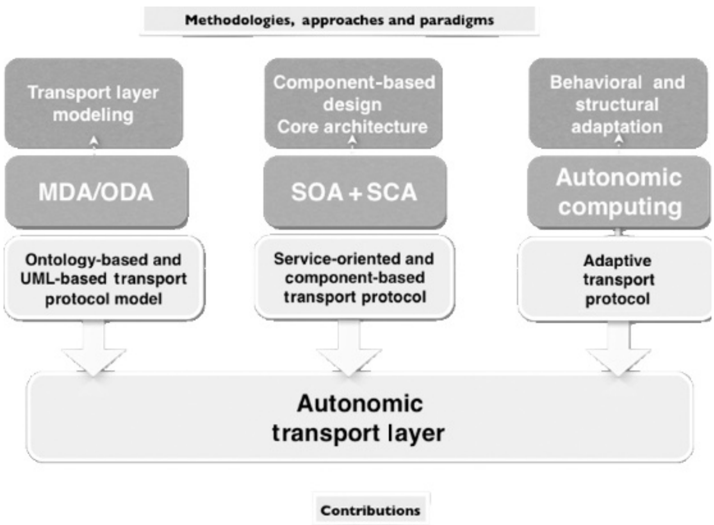

Figure 1.2. *Book structure*

This book is structured as follows: Chapters 2 and 3 (transport layer modeling) present the state of the art on transport services and protocols and propose a first contribution represented by an ontology-based transport protocol model integrating the semantic related to services, protocols, functions and mechanisms. Likewise, a UML-based transport protocol model as well as an MDA approach are proposed for the incremental and iterative design of the next-generation autonomic transport layer (ATL) in Chapters 4, 5 and 6.

Service-oriented and component-based design: Chapter 8 presents the service-oriented and component-based transport protocol. Chapter 9 includes a state of the art on service-oriented and service-component architectures and describes how these approaches can guide the design of the next-generation transport layer architecture.

Adaptive and autonomic properties: Chapter 10 presents the adaptive transport protocol. This protocol integrates adaptive strategies aimed at implementing behavioral and structural adaptation actions based on the network environment conditions and guided by the application requirements. Chapter 11 presents the design principles guiding the design of an autonomic transport protocol.

Finally, based on the lessons learned from the introduced model-driven, service-oriented and component-based approaches and the benefits offered by the autonomic computing paradigm, the design principles and guidelines aimed at driving the design and the development of the next-generation transport layer are presented.