Prototyping with the Future Internet Toolbox*

C. Dannewitz, T. Biermann, M. Dräxler, F. Beister, and H. Karl

University of Paderborn, Research Group Computer Networks
{cdannewitz,thobie,draexler,azamir,hkarl}@mail.upb.de

Abstract. Prototyping future Internet technologies is an important but complicated task. To ease this process, we have developed the *Future Internet Toolbox (FIT)*. In this paper, we demonstrate how it facilitates prototyping in practice by presenting three scenarios that have been implemented using *FIT*.

1 Introduction

Currently, a lot of research focuses on future Internet technologies. To simplify the process of prototyping and evaluating new concepts and protocols, we developed the Future Internet Toolbox (FIT) – a collection of 4 frameworks covering information-centric networking, data transport, naming, and name resolution.

We have introduced FIT in detail in [1,2]. This paper now focuses on three specific use cases where FIT helped us prototyping new networking features: efficient and secure information-centric data dissemination (Sec. 3.1), rapid information-centric application development (Sec. 3.2), and session mobility for video streaming (Sec. 3.3). We shortly revisit the FIT frameworks in Sec. 2.

2 FIT Frameworks

2.1 Information-Centric Networking and Secure Naming

Several Information-Centric Network (ICN) architectures have been proposed lately (see [2] for details) to overcome the inefficiencies of today's Internet architecture with respect to information dissemination and information handling. The ICN framework supports and accelerates the design and evaluation of ICN architectures. It provides three main aspects: (1) a generic, adaptable *node structure* for building custom information-centric nodes, (2) implementations of various *components* that can be used to compose such a node, and (3) a testbed for interconnecting those nodes into an overall ICN architecture.

The ICN framework is closely connected to the naming framework. The naming framework supports a wide variety of naming schemes, including complex ones involving security-related features like data integrity checking. At the same time, it minimizes the implementation overhead of new naming schemes.

* The research leading to these results has been conducted in the 4WARD project and received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 216041.

T. Magedanz et al. (Eds.): TridentCom 2010, LNICST 46, pp. 609–611, 2011.

2.2 Data Transport and Name Resolution

The data transport framework bases on the Generic Path (GP) architecture that has been developed in the European research project 4WARD. This architecture's goal is to overcome limitations of the ISO/OSI architecture, which is mainly the difficulty to introduce new functionality into today's networks.

This has been achieved by designing the framework for flexibility and extensibility from the beginning. It is service-oriented, i.e., Entities ("layers") can be arbitrarily composed based on their required and provided services (no static stack determined by the technologies anymore). Another key issue are the unified structures. Entities and Endpoints (the termination of a communication path) all have the same interfaces to ease cross-layer networking. Additionally, reusable components like Entities and Endpoints, or mechanisms like name resolution are made explicit to speed up developing new features. As a result, name resolution uses the same mechanisms on all different levels of networking.

3 Use Cases

3.1 Efficient and Secure Information-Centric Data Dissemination

In this scenario, we demonstrate efficient and secure data dissemination mechanisms for the future Internet, based on the FIT-based Network of Information (NetInf) prototype [1]. We used the ICN framework to build information-centric NetInf nodes, both infrastructure and client nodes. Each NetInf node can cache and provide data to other NetInf nodes. In addition, they provide a name resolution service to resolve flat NetInf names into download locations.

A Web browser, extended via a NetInf plugin to allow communicating with the NetInf infrastructure, handles links to flat names instead of URLs by triggering a name resolution. During resolution, an appropriate download location is selected by the NetInf nodes; the browser downloads and displays the requested information. In addition, the client NetInf node checks the data's integrity using the FIT naming framework and caches the data locally. This enables to provide the data to other nodes to increase availability in case of network disruption and to reduce Internet traffic. The whole process is transparent to the user.

3.2 Rapid Information-Centric Application Development

Building information-centric applications involves many similar components like information model, distributed storage, efficient and secure data dissemination, and notification services. Building such applications and the required infrastructure on a global scale based on today's Internet architecture is often cumbersome as most components have to be rebuilt and/or reintegrated for each project.

This scenario illustrates how even complex information-centric applications and the corresponding infrastructure can easily be developed using building blocks (*Information Model*, *Naming Scheme*, *Event/Storage/Search Service*) of the FIT-based NetInf prototype. These blocks enabled us to rapidly develop a globally scalable collaborative editing and context-awareness application. Specifically, we developed a shopping application that informs the user about available products in near by shops from his shopping list. The shopping list can be edited collaboratively by multiple users. This is implemented by storing the data in a NetInf-specific data structure and utilizes the Event Service to subscribe the applications and a specialized Search Service to any changes performed on the shopping lists, inventory lists, and the user's geo-locations.

3.3 Session Mobility for Video Streaming

In this scenario we focus on FIT's data transport part and show how session mobility can be implemented for a live video stream, i.e., client and server are moved to different machines during runtime.

For this, we first added support for session mobility to the `AbstractEntity`. This covers functions to serialize the Endpoint of an existing GP and to transfer this serialization to another Entity. The GP management functions have been extended to relocate the GP alongside the Endpoint transfer to the new Entity. All these mechanisms are generic, i.e., they can be used for any GP type.

Second, we created a `VideoEndpoint` that is instantiated by a `VideoEntity`. This Endpoint uses the VLC media player to generate and display video streams. In particular, the `VideoEndpoint` contains the current state of VLC (playback position and source video file), which is eventually contained in the Endpoint serialization. This permits session mobility by relocating the video Endpoints on both client and server side. The server-side session mobility is especially useful for dynamic resource allocation for single video streams; a problem that cannot be solved by server migration using virtualization.

4 Conclusion

Experience from implementing the three scenarios with the FIT frameworks shows that FIT simplifies prototyping significantly. Due to its generality, it is useful for many different projects as it reduces redundant implementation of basic testbed functions and provides ready-to-use building blocks. This way, new, small components can be complemented with an overall network architecture.

References

1. Biermann, T., Dannewitz, C., Karl, H.: FIT: Future Internet Toolbox. In: Magedanz, T., et al. (eds.) Tridentcom 2010. LNICST, vol. 46, pp. 444–453. Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (2010)
2. Biermann, T., Dannewitz, C., Karl, H.: FIT: Future Internet Toolbox — extended report. Technical Report TR-RI-10-311, University of Paderborn (February 2010)